**CTDL097 - Tiền tố về dạng hậu tố**

#include <bits/stdc++.h>

using namespace std;

string ts = "+-\*/^";

bool check(char c)

{

for (int i = 0; i < ts.size(); i++)

if (ts[i] == c)

return 1;

return 0;

}

bool Var(char c)

{

return ((c >= 'A' && c <= 'Z') || (c >= 'a' && c <= 'z'));

}

int main()

{

int t;

cin >> t;

while (t--)

{

string s, k, ans;

cin >> s;

stack<string> st;

for (int i = 0; i < s.size(); i++)

{

if (Var(s[i]))

st.push(string(1, s[i]));

if (check(s[i]))

{

k = st.top();

st.pop();

k = s[i] + k;

k = st.top() + k;

st.pop();

k = "(" + k + ")";

st.push(k);

}

}

s = st.top();

while (!st.empty())

st.pop();

for (int i = s.size() - 1; i >= 0; i--)

{

if (Var(s[i]))

ans = s[i] + ans;

if (check(s[i]))

st.push(string(1, s[i]));

if (s[i] == '(')

{

ans = st.top() + ans;

st.pop();

}

}

while (!st.empty())

{

ans = st.top() + ans;

st.pop();

}

cout << ans << endl;

}

}

## CTDL001 - Sinh xâu nhị phân N bit

#include <bits/stdc++.h>

using namespace std;

int main()

{

int t;

cin >> t;

while (t--){

int n;

cin >>n;

int a[n+5];

for(int i = 1; i <= n; ++i) a[i] = 0;

int ok = 1;

while(ok){

for(int i = 1; i <= n; ++i) cout <<a[i] <<' ';

int i = n;

while(i > 0 && a[i] == 1) a[i] = 0, i--;

if(i == 0) ok = 0;

else a[i] = 1;

cout <<'\n';

}

}

return 0;

}

## CTDL002 - chuỗi nhị phân theo hợp lệ

#include <iostream>

#include <string>

using namespace std;

void generateAllStrings(string& s, int index)

{

if (index == s.size())

{

cout << s << endl;

return;

}

if (s[index] == '?')

{

s[index] = '0';

generateAllStrings(s, index + 1);

s[index] = '1';

generateAllStrings(s, index + 1);

s[index] = '?';

}

else

{

generateAllStrings(s, index + 1);

}

}

int main()

{

int t;

cin >> t;

while (t--)

{

string s;

cin >> s;

generateAllStrings(s, 0);

}

return 0;

}

## CTDL003 - Xâu nhị phân kế tiếp

#include <bits/stdc++.h>

using namespace std;

string bin;

void numnext () {

int l=bin.length(), check=1, k;

for (int i=l-1; i>=0; i--)

if (bin[i]=='0') {

check=0;

k=i;

break;

}

bin[k]='1';

for (int i=k+1; i<=l; i++)

bin[i]='0';

if (check) {

for (int i=0; i<l; i++)

cout <<0;

} else {

cout <<bin;

}

cout <<endl;

}

int main () {

int t;

cin >>t;

while (t--) {

cin >>bin;

numnext ();

}

return 0;

}

## CTDL004 - Sinh xâu AB

#include <iostream>

#include <string>

using namespace std;

void generate\_AB\_strings(string current, int n) {

if (current.length() == n) {

cout << current << " ";

} else {

generate\_AB\_strings(current + "A", n);

generate\_AB\_strings(current + "B", n);

}

}

int main() {

int T;

cin >> T;

for (int t = 0; t < T; t++) {

int n;

cin >> n;

string current = "";

generate\_AB\_strings(current, n);

cout << endl;

}

return 0;}

## CTDL005 - Sinh hoán vị

#include <iostream>

#include <vector>

using namespace std;

void generatePermutations(vector<int>& current, vector<bool>& used, int n) {

if (current.size() == n) {

for (int i = 0; i < n; i++) {

cout << current[i];

}

cout << " ";

return;

}

for (int i = 1; i <= n; i++) {

if (!used[i]) {

current.push\_back(i);

used[i] = true;

generatePermutations(current, used, n);

current.pop\_back();

used[i] = false;

}

}

}

int main() {

int T;

cin >> T;

for (int t = 0; t < T; t++) {

int n;

cin >> n;

vector<int> current;

vector<bool> used(n + 1, false);

generatePermutations(current, used, n);

cout << endl;

}

return 0;

}

## CTDL007 - Hoán vị ngược

#include <bits/stdc++.h>

using namespace std;

int main()

{

int t;

cin >> t;

while (t--)

{

int n;

cin >> n;

int a[n + 1];

for (int i = 1; i <= n; i++)

a[i] = i;

stack<string> st;

while (1)

{

string s = "";

for (int i = 1; i <= n; i++)

s += to\_string(a[i]);

st.push(s);

int ok = 0;

for (int i = n - 1; i >= 1; i--)

{

if (a[i] < a[i + 1])

{

ok = 1;

sort(a + i + 1, a + n + 1);

for (int j = i + 1; j <= n; j++)

{

if (a[j] > a[i])

{

swap(a[i], a[j]);

break;

}

}

break;

}

}

if (ok == 0)

break;

}

while (st.size())

{

cout << st.top() << " ";

st.pop();

}

cout << endl;

}

}

## CTDL008 - Mã gray 002

#include <bits/stdc++.h>

using namespace std;

string cvt(string s){

string ans ="";

ans += s[0];

for(int i = 1; i < s.size(); ++i){

ans += (s[i] != s[i-1]) ? '1': '0';

}

return ans;

}

int main()

{

int t;

cin >> t;

while (t--){

string s;

cin >>s;

cout <<cvt(s) <<'\n';

}

return 0;

}

## CTDL011 - Chuỗi nhị phân có bit 01 thỏa mã điều kiện

#include<bits/stdc++.h>

using namespace std;

#define int long long

signed main(){

int t;

cin >> t;

while(t--){

int n;

cin >> n;

if(n < 4) cout << -1 ;

else {

for(int i = 0 ; i < (1 << n) ; i ++){

string s = bitset<32>(i).to\_string();

s = s.substr(32-n);

size\_t first = s.find("01");

size\_t second = s.find("01",first + 2);

if(first != string::npos && second != string::npos && s.find("01",second + 2) == string :: npos) cout << s << " ";

}

cout << endl;

}

}

return 0;

}

## CTDL012 - Bội số

#include <bits/stdc++.h>

using namespace std;

int main()

{

ios\_base::sync\_with\_stdio(0);

cin.tie(0);

int t;

cin >> t;

while (t--)

{

int n;

cin >> n;

queue<long long> q;

q.push(9);

while (q.size())

{

long long k = q.front();

q.pop();

if (k % n == 0)

{

cout << k << endl;

break;

}

q.push(k \* 10);

q.push(k \* 10 + 9);

}

}

}

## CTDL013 - Tổ hợp

#include <bits/stdc++.h>

using namespace std;

int main()

{

int t;

cin >> t;

while (t--)

{

int n, k;

cin >> n >> k;

int a[k + 1];

for (int i = 1; i <= k; i++)

a[i] = i;

while (1)

{

for (int i = 1; i <= k; i++)

cout << a[i];

cout << " ";

int ok = 0;

for (int i = k; i >= 1; i--)

{

if (a[i] != n - k + i)

{

ok = 1;

a[i]++;

for (int j = i + 1; j <= k; j++)

a[j] = a[j - 1] + 1;

break;

}

}

if (ok == 0)

break;

}

cout << endl;

}

}

## CTDL019 - HOÁN VỊ KẾ TIẾP

#include <bits/stdc++.h>

using namespace std;

int main()

{

int t;

cin >>t;

while(t--){

int n;

cin >>n;

int a[n];

for(int i = 0; i < n; ++i) cin >>a[i];

if(next\_permutation(a,a+n)) for(int i = 0; i < n; ++i) cout <<a[i] <<' ';

else for(int i = 0; i < n; ++i) cout <<i+1 <<' ';

cout <<'\n';

}

return 0;

}

## CTDL018 - Biến đổi dãy số 1

#include <iostream>

#include <vector>

using namespace std;

vector<vector<int>> generateSumTriangle(const vector<int>& A) {

int n = A.size();

vector<vector<int>> triangle(n);

triangle[0] = A;

for (int i = 1; i < n; i++) {

for (int j = 0; j < n - i; j++) {

triangle[i].push\_back(triangle[i - 1][j] + triangle[i - 1][j + 1]);

}

}

return triangle;

}

int main() {

int T;

cin >> T;

while (T--) {

int N;

cin >> N;

vector<int> A(N);

for (int i = 0; i < N; i++) {

cin >> A[i];

}

vector<vector<int>> triangle = generateSumTriangle(A);

for (const auto& row : triangle) {

cout << "[";

for (int i = 0; i < row.size(); i++) {

cout << row[i];

if (i != row.size() - 1) {

cout << " ";

}

}

cout << "]" << endl;

}

}

return 0;

}

## CTDL017 - Biến đổi dãy số 2

#include <iostream>

#include <vector>

using namespace std;

vector<vector<int>> generateSpecialTriangle(vector<int> A) {

vector<vector<int>> triangle;

triangle.push\_back(A);

while (A.size() > 1) {

vector<int> row;

for (int i = 0; i < A.size() - 1; i++) {

row.push\_back(A[i] + A[i + 1]);

}

triangle.push\_back(row);

A = row;

}

return triangle;

}

int main() {

int T;

cin >> T;

while (T--) {

int N;

cin >> N;

vector<int> A(N);

for (int i = 0; i < N; i++) {

cin >> A[i];

}

vector<vector<int>> triangle = generateSpecialTriangle(A);

for (const auto& row : triangle) {

cout << "[";

for (int i = 0; i < row.size(); i++) {

cout << row[i];

if (i < row.size() - 1) {

cout << " ";

}

}

cout << "]" << endl;

}

}

return 0;

}

## CTDL016 - Lũy thừa ma trận 1

#include <bits/stdc++.h>

using namespace std;

struct data

{

long long x[10][10];

};

int n;

long long mod = 1000000007;

struct data operator\*(struct data a, struct data b)

{

struct data xx;

for (int i = 0; i < n; i++)

{

for (int j = 0; j < n; j++)

{

xx.x[i][j] = 0;

for (int k = 0; k < n; k++)

xx.x[i][j] = (xx.x[i][j] + (a.x[i][k] \* b.x[k][j]) % mod) % mod;

}

}

return xx;

};

struct data poww(struct data a, long long k)

{

if (k == 1)

return a;

if (k % 2 == 1)

return a \* poww(a, k - 1);

struct data xx = poww(a, k / 2);

return xx \* xx;

}

int main(){

int t;

cin >> t;

while (t--)

{

long long k;

cin >> n >> k;

struct data a;

for (int i = 0; i < n; i++)

for (int j = 0; j < n; j++)

cin >> a.x[i][j];

a = poww(a, k);

unsigned long long ans=0;

for (int i = 0; i < n; i++)

{

ans += a.x[i][i];

ans %= mod;

*//cout << a.x[i][j] << " ";*

}

cout<<ans << '\n';

}

}

## CTDL014 - Lũy thừa ma trận 2

#include <bits/stdc++.h>

const int MOD=1e9+7;

using namespace std;

struct data

{

long long x[10][10];

};

int n;

long long mod = 1000000007;

struct data operator\*(struct data a, struct data b)

{

struct data xx;

for (int i = 0; i < n; i++)

{

for (int j = 0; j < n; j++)

{

xx.x[i][j] = 0;

for (int k = 0; k < n; k++)

xx.x[i][j] = (xx.x[i][j] + (a.x[i][k] \* b.x[k][j]) % mod) % mod;

}

}

return xx;

};

struct data poww(struct data a, long long k)

{

if (k == 1)

return a;

if (k % 2 == 1)

return a \* poww(a, k - 1);

struct data xx = poww(a, k / 2);

return xx \* xx;

}

int main()

{

int t;

cin >> t;

while (t--)

{

long long k;

cin >> n >> k;

struct data a;

for (int i = 0; i < n; i++)

for (int j = 0; j < n; j++)

cin >> a.x[i][j];

a = poww(a, k);

unsigned long long ans=0;

for (int i = 0; i < n; i++)

{

ans += a.x[n - i - 1][i];

ans %= mod;

}

cout<<ans << '\n';

}

}

## CTDL030 - Đổi chỗ xâu ký tự

#include <bits/stdc++.h>

using namespace std;

int main()

{

int t;

cin >> t;

while (t--){

string s;

int k;

cin >> k >>s;

int n = s.size();

for(int i=0;i<=n-1;i++){

char l = s[n-1];

int vt = n-1;

for(int j = n-1; j > i && k > 0; --j){

if(l < s[j]) l = s[j], vt=j;

}

if(l > s[i] && k > 0) swap(s[i],s[vt]),k--;

}

cout <<s <<'\n';

}

return 0;

}

## CTDL031 - Mãy Gray004

#include <iostream>

#include <bitset>

using namespace std;

int grayCode(int n) {

return n ^ (n >> 1);

}

int main() {

int T;

cin >> T;

while (T--) {

int N;

cin >> N;

int gray = grayCode(N);

cout << gray << endl;

}

return 0;

}

## CTDL032 - Xâu nhị phân có độ dài N

#include <cstdio>

#include<bits/stdc++.h>

#define mod 1000000000

using namespace std;

int f[1111111],n,k;

int main(){

int t;

cin>>t;

while(t--){

memset(f,0,sizeof(f));

scanf("%d %d",&n,&k);

f[0] = 2; f[1] = 2;

for(int i = 2;i<=n;i++){

if(i<=k) f[i] = (f[i-1]\*2)%mod;

else f[i] = (f[i-1]\*2-f[i-k-1])%mod;

if(f[i]<0) f[i]+=mod;

}

printf("%d\n",f[n]);

}

}

## CTDL033 - TÌM SỐ NGUYÊN THỎA MÃN ĐIỀU KIỆN

#include <bits/stdc++.h>

using namespace std;

#define int long long

int bezout(int a, int b) {

int xa = 1, xb = 0;

while (b) {

int q = a / b;

int r = a - q \* b, xr = xa - q \* xb;

a = b; xa = xb;

b = r; xb = xr;

}

return xa;

}

int solve(int a, int b, int c) {

int g = \_\_gcd(a, b);

int x = bezout(a, b) \* (c / g);

int p = abs(b / g), q = abs(a / g);

x = (x % p + p) % p;

int y = (c - a \* x) / b;

int cur = x + y;

if (x < 0 || y < 0) cur = -1;

int yy = (y % q + q) % q;

int xx = (c - b \* yy) / a;

if (cur == -1 || (min(xx, yy) >= 0 && cur > xx + yy))

cout << xx << ' ' << yy << endl;

else

cout << x << ' ' << y << endl;

}

main() {

int t;

cin>>t;

while(t--){

int a1, a2, b1, b2;

cin >> a1 >> b1 >> a2 >> b2;

solve(a1, -a2, b2 - b1);

}

}

## CTDL035 - Lũy Thừa 001

#include <bits/stdc++.h>

using namespace std;

typedef long double ld;

const ld epsilon = 1e-6;

int main()

{

int test;

cin >> test;

while (test--)

{

ld a, b;

int ans;

cin >> a >> b;

for (int k = 1; k <= 40; k++)

{

ld l = pow(a, 1.0 / k);

ld r = pow(b, 1.0 / k);

ld checkleft = l - floor(l);

ld checkright = r - ceil(r);

if (abs(checkleft) <= epsilon)

l = floor(l);

else

l = ceil(l);

if (abs(checkright) <= epsilon)

r = ceil(r);

else

r = floor(r);

if (r - l >= 0)

{

ans = k;

}

}

cout << ans << endl;

}

return 0;

}

## CTDL036 - KIỂM TRA ĐỒ THỊ CÓ PHẢI LÀ CÂY HAY KHÔNG

#include<bits/stdc++.h>

using namespace std;

typedef long long ll;

const int mxN = 1001;

vector<int> adj[mxN];

int V, E, n;

bool flag;

bool visited[mxN];

int sz[mxN];

int par[mxN];

void make\_set(){

for(int i = 1; i <= n; ++i){

sz[i] = 1;

par[i] = i;

}

}

int Find(int u){

if(u == par[u]) return u;

return par[u] = Find(par[u]);

}

bool Union(int x, int y){

x = Find(x);

y = Find(y);

if(x == y) return false;

if(sz[x] < sz[y]) swap(x,y);

sz[x] += y;

par[y] = x;

return true;

}

int main(){

ios\_base::sync\_with\_stdio(false);

cin.tie(0);

int t;

cin >> t;

while(t--){

flag = true;

cin >> n;

make\_set();

for(int i = 0; i < n - 1; ++i){

int u, v;

cin >> u >> v;

if(!Union(u,v)) flag = false;

}

if(flag)

cout << "YES";

else

cout << "NO";

cout << '\n';

}

return 0;

}

## CTDL037 - Đổi tiền

#include <bits/stdc++.h>

using namespace std;

int main()

{

int t;

cin >> t;

int a[100001] = {};

int b[10] = {1, 2, 5, 10, 20, 50, 100, 200, 500, 1000};

for (int i = 1; i <= 100000; i++)

{

a[i] = 1e9;

for (int j = 0; j < 10; j++)

{

if (i == b[j])

{

a[i] = 1;

break;

}

if (i - b[j] > 0 && a[i - b[j]] > 0)

a[i] = min(a[i], a[i - b[j]] + 1);

}

}

while (t--)

{

int n;

cin >> n;

cout << a[n] << endl;

}

}

## CTDL038 - Nhầm số

#include <bits/stdc++.h>

const int MOD=1e9+7;

using namespace std;

int main(){

int t;

cin >>t;

while(t--){

string a,b;

cin.ignore();

cin>>a>>b;

int c1=0,d1=0,c2=0,d2=0;

for(int i=0;i<a.length();i++){

if(a[i]=='3') a[i]='5';

c1=c1\*10+a[i]-48;

}

for(int i=0;i<b.length();i++){

if(b[i]=='3') b[i]='5';

c2=c2\*10+b[i]-48;

}

c1+=c2;

for(int i=0;i<a.length();i++){

if(a[i]=='5') a[i]='3';

d1=d1\*10+a[i]-48;

}

for(int i=0;i<b.length();i++){

if(b[i]=='5') b[i]='3';

d2=d2\*10+b[i]-48;

}

d1+=d2;

cout<<d1<<" "<<c1<<'\n';

}

return 0;

}

## CTDL039 - Tìm max

#include <bits/stdc++.h>

using namespace std;

int main()

{

int t;

cin >> t;

while (t--)

{

int n;

cin >> n;

long long a[n], s = 0, mod = 1e9 + 7;

for (int i = 0; i < n; i++)

cin >> a[i];

sort(a, a + n);

for (int i = 0; i < n; i++)

s = (s + a[i] \* i) % mod;

cout << s << endl;

}

}

## CTDL040 - Tổng nhỏ nhất

#include <bits/stdc++.h>

using namespace std;

int main()

{

int t;

cin >> t;

while (t--)

{

int n;

cin >> n;

int a[n];

long long s1 = 0, s2 = 0;

for (int i = 0; i < n; i++)

cin >> a[i];

sort(a, a + n);

for (int i = 0; i < n; i += 2)

s1 = s1 \* 10 + a[i];

for (int i = 1; i < n; i += 2)

s2 = s2 \* 10 + a[i];

cout << s1 + s2 << endl;

}

}

## CTDL041 - Chuyển từ danh sách cạnh sang danh sách kề

#include <iostream>

#include <vector>

#include <algorithm>

using namespace std;

void printAdjacencyList(vector<vector<int>>& adjacencyList) {

int numVertices = adjacencyList.size();

for (int i = 0; i < numVertices; i++) {

cout << "D" << i + 1 << ": ";

for (int j = 0; j < adjacencyList[i].size(); j++) {

cout << adjacencyList[i][j] << " ";

}

cout << endl;

}

}

int main() {

int T;

cin >> T;

while (T--) {

int V, E;

cin >> V >> E;

vector<vector<int>> adjacencyList(V);

for (int i = 0; i < E; i++) {

int u, v;

cin >> u >> v;

adjacencyList[u - 1].push\_back(v);

adjacencyList[v - 1].push\_back(u);

}

printAdjacencyList(adjacencyList);

}

return 0;

}

## CTDL042 - Giá trị nhỏ nhất cảu biểu thức

#include<bits/stdc++.h>

using namespace std;

typedef long long ll;

const ll mod = 1e9 + 7;

int main(){

int t;

cin >> t;

while(t--){

int n;

cin >> n;

vector<ll> a1(n), a2(n);

for(int i = 0; i < n; ++i)

cin >> a1[i];

for(int i = 0; i < n; ++i)

cin >> a2[i];

sort(a1.begin(), a1.end());

sort(a2.begin(), a2.end(), greater<ll>());

ll res = 0;

for(int i = 0; i < n; ++i){

res += a1[i] \* a2[i];

}

cout << res << '\n';

}

return 0;

}

## CTDL043 - Sắp xếp công việc 1

#include <iostream>

#include <vector>

#include <algorithm>

int findMaxActions(std::vector<int>& startTimes, std::vector<int>& endTimes) {

int n = startTimes.size();

std::vector<std::pair<int, int>> actions;

for (int i = 0; i < n; i++) {

actions.push\_back({startTimes[i], endTimes[i]});

}

std::sort(actions.begin(), actions.end(), [](const std::pair<int, int>& a, const std::pair<int, int>& b) {

return a.second < b.second;

});

int count = 1;

int endTime = actions[0].second;

for (int i = 1; i < n; i++) {

if (actions[i].first >= endTime) {

count++;

endTime = actions[i].second;

}

}

return count;

}

int main() {

int t;

std::cin >> t;

while (t--) {

int n;

std::cin >> n;

std::vector<int> startTimes(n);

std::vector<int> endTimes(n);

for (int i = 0; i < n; i++) {

std::cin >> startTimes[i];

}

for (int i = 0; i < n; i++) {

std::cin >> endTimes[i];

}

int maxActions = findMaxActions(startTimes, endTimes);

std::cout << maxActions << std::endl;

}

return 0;

}

## CTDL044 - Nối dây 1

#include <bits/stdc++.h>

using namespace std;

int main()

{

ios\_base::sync\_with\_stdio(0);

cin.tie(0);

int t;

cin >> t;

while (t--)

{

int n, x;

cin >> n;

long long s = 0;

priority\_queue<int, vector<int>, greater<int>> q;

for (int i = 0; i < n; i++)

{

cin >> x;

q.push(x);

}

while (q.size() > 1)

{

int s1 = q.top();

q.pop();

int s2 = q.top();

q.pop();

int k = s1 + s2;

s += k;

q.push(k);

}

cout << s << endl;

}

}

## CTDL045 - SẮP ĐẶT XÂU KÝ TỰ 1

#include <bits/stdc++.h>

#define endl "\n"

using namespace std;

int main()

{

int t;

cin >> t;

while (t--)

{

string s;

cin >> s;

int a[30] = {}, MAX = 0;

for (int i = 0; i < s.size(); i++)

a[s[i] - 'a']++;

for (int i = 0; i < 30; i++)

MAX = max(MAX, a[i]);

if (MAX \* 2 > s.size() + 1)

cout << -1 << endl;

else

cout << 1 << endl;

}

}

## CTDL046 - Mua lương thực

#include <bits/stdc++.h>

using namespace std;

int main()

{

ios\_base::sync\_with\_stdio(0);

cin.tie(0);

int t;

cin >> t;

while (t--)

{

int n, s, m;

cin >> n >> s >> m;

if (s \* m > (s - s / 7) \* n)

cout << -1 << endl;

else

{

for (int i = 1; i <= s - s / 7; i++)

{

if (n \* i >= s \* m)

{

cout << i << endl;

break;

}

}

}

}

}

## CTDL047 - SỐ MAY MẮN

#include <bits/stdc++.h>

using namespace std;

int main()

{

ios\_base::sync\_with\_stdio(0);

cin.tie(0);

int t;

cin >> t;

while (t--)

{

int n, ok = 0;

cin >> n;

for (int i = 0; i <= n / 4; i++)

{

if ((n - i \* 4) % 7 == 0)

{

ok = 1;

for (int j = 0; j < i; j++)

cout << 4;

for (int j = 0; j < (n - i \* 4) / 7; j++)

cout << 7;

cout << endl;

break;

}

}

if (ok == 0)

cout << -1 << endl;

}

}

## CTDL048 - PHÂN SỐ ĐƠN VỊ

#include <bits/stdc++.h>

using namespace std;

int main()

{

ios\_base::sync\_with\_stdio(0);

cin.tie(0);

int t;

cin >> t;

while (t--)

{

long long a, b, n, m, x;

cin >> a >> b;

while (b % a != 0)

{

n = a;

m = b;

x = b / a;

cout << 1 << "/" << x + 1 << " + ";

a = (x + 1) \* n - m;

b = (x + 1) \* b;

}

cout << 1 << "/" << b / a;

cout << endl;

}

}

## CTDL049 - TÌM KIẾM NHỊ PHÂN

#include <bits/stdc++.h>

#define endl "\n"

using namespace std;

int main()

{

ios\_base::sync\_with\_stdio(0);

cin.tie(0);

int t;

cin >> t;

while (t--)

{

int n, k;

cin >> n >> k;

int a[n];

for (int i = 0; i < n; i++)

{

cin >> a[i];

}

int m = lower\_bound(a, a + n, k) - a;

if (a[m] != k)

cout << "NO" << endl;

else

cout << m + 1 << endl;

}

}

## CON4\_24 - ĐẾM DÃY

#include<bits/stdc++.h>

using namespace std;

typedef long long ll;

const ll mod = 123456789;

ll Pow(ll n, ll k){

ll res = 1;

while(k){

if(k&1){

res \*= n;

res %= mod;

}

n \*= n;

n %= mod;

k >>= 1;

}

return res;

}

int main(){

int t;

cin >> t;

while(t--){

ll n;

cin >> n;

cout << Pow(2, n - 1) << '\n';

}

return 0;

}

## CTDL052 - HỆ CƠ SỐ K

#include <bits/stdc++.h>

const int MOD=1e9+7;

using namespace std;

int convertToDecimal(string num, int K) {

int decimal = 0;

int power = 1;

for (int i = num.length() - 1; i >= 0; i--) {

int digit = 0;

if (num[i] >= '0' && num[i] <= '9') {

digit = num[i] - '0';

} else {

digit = num[i] - 'A' + 10;

}

decimal += digit \* power;

power \*= K;

}

return decimal;

}

string convertFromDecimal(int num, int K) {

string result = "";

while (num > 0) {

int digit = num % K;

if (digit < 10) {

result = char(digit + '0') + result;

} else {

result = char(digit - 10 + 'A') + result;

}

num /= K;

}

return result;

}

string add(string A, string B, int K) {

int decimalA = convertToDecimal(A, K);

int decimalB = convertToDecimal(B, K);

int sum = decimalA + decimalB;

string result = convertFromDecimal(sum, K);

return result;

}

int main(){

int t;

cin >> t;

while(t--){

int k;

string a, b;

cin >>k >>a >>b;

cout <<add(a,b,k) <<'\n';

}

return 0;

## }

## CON4\_27\_CTDL053 - ĐẾM SỐ BIT 1

#include <bits/stdc++.h>

using namespace std;

long long Find(long long pos, long long n, long long ctr){

if (pos & 1)

return 1;

if (pos == ctr)

return (n % 2);

if (pos > ctr)

return Find(pos - ctr, n / 2, ctr / 2);

return Find(pos, n / 2, ctr / 2);

}

int main(){

int t;

cin >> t;

while (t--)

{

long long n, l, r;

cin >> n >> l >> r;

long long ctr = pow(2, (long long)log2(n)), ans = 0;

for (long long i = l; i <= r; i++)

ans += Find(i, n, ctr);

cout << ans << "\n";

}

## }

## CTDL054 - CẶP ĐIỂM GẦN NHẤT

#include <iostream>

#include <vector>

#include <cmath>

#include <iomanip>

using namespace std;

double distance(int x1, int y1, int x2, int y2) {

return sqrt(pow(x2 - x1, 2) + pow(y2 - y1, 2));

}

int main() {

int T, N;

cin >> T;

while (T--) {

cin >> N;

vector<pair<int, int>> points(N);

for (int i = 0; i < N; ++i) {

cin >> points[i].first >> points[i].second;

}

double minDist = 1e9;

for (int i = 0; i < N; ++i) {

for (int j = i + 1; j < N; ++j) {

double dist = distance(points[i].first, points[i].second, points[j].first, points[j].second);

if (dist < minDist) {

minDist = dist;

}

}

}

cout << fixed << setprecision(6) << minDist << " "<< endl;

}

return 0;

## }

## CON4\_30 - SỐ FIBONACCI THỨ N

#include <bits/stdc++.h>

using namespace std;

long long mod = 1e9 + 7, F[2][2], M[2][2];

void Mul(long long f[2][2], long long m[2][2])

{

long long x = (f[0][0] \* m[0][0] % mod + f[0][1] \* m[1][0] % mod) % mod;

long long y = (f[0][0] \* m[0][1] % mod + f[0][1] \* m[1][1] % mod) % mod;

long long z = (f[1][0] \* m[0][0] % mod + f[1][1] \* m[1][0] % mod) % mod;

long long t = (f[1][0] \* m[0][1] % mod + f[1][1] \* m[1][1] % mod) % mod;

F[0][0] = x;

F[0][1] = y;

F[1][0] = z;

F[1][1] = t;

}

void Pow(long long f[2][2], long long n)

{

if (n <= 1)

return;

Pow(f, n / 2);

Mul(f, f);

if (n & 1)

Mul(f, M);

}

long long fibo(long long n)

{

F[0][0] = F[0][1] = F[1][0] = 1;

F[1][1] = 0;

M[0][0] = M[0][1] = M[1][0] = 1;

M[1][1] = 0;

Pow(F, n - 1);

return F[0][0];

}

int main()

{

int t;

cin >> t;

while (t--)

{

long long n;

cin >> n;

cout << fibo(n) << endl;

}

## }

## CTDL056 - Tích nhị phân

#include<bits/stdc++.h>

using namespace std;

typedef long long ll;

int main()

{

ios\_base::sync\_with\_stdio(false);

cin.tie(0);

int t;

cin >> t;

while(t--){

string s1 ,s2;

cin >> s1 >> s2;

ll n1 = 0, n2 = 0;

ll tmp = 1;

for(int i = s1.length() - 1; ~i; --i){

if(s1[i] == '1')

n1 += tmp;

tmp <<= 1;

}

tmp = 1;

for(int i = s2.length() - 1; ~i; --i){

if(s2[i] == '1')

n2 += tmp;

tmp <<= 1;

}

cout << n1 \* n2 << '\n';

}

## }

## CON4\_37 - Tính FloorX()

#include <bits/stdc++.h>

#define ll long long

#define f(i,a,b) for(ll i=a; i<=b; ++i)

using namespace std;

ll findx(ll a[], ll n, ll k){

ll l = 0, r =n-1;

ll res = -2;

while(l <= r){

ll mid = l +(r-l)/2;

if(a[mid] <= k){

res = mid;

l = mid+1;

}

else{

r = mid-1;

}

}

return res + 1;

}

int main(){

int t;

cin >>t;

while(t--){

ll n, k;

cin >>n >>k;

ll a[n];

f(i,0,n-1) cin >>a[i];

sort(a,a+n);

cout <<findx(a,n,k) <<'\n';

}

return 0;

## }

## CTDL059 - PHẦN TỬ KHÁC NHAU.

#include <bits/stdc++.h>

#define endl "\n"

using namespace std;

int main()

{

int t;

cin >> t;

while (t--)

{

int n;

cin >> n;

int a[n], b[n];

for (int i = 0; i < n; i++)

cin >> a[i];

for (int i = 0; i < n - 1; i++)

cin >> b[i];

for (int i = 0; i < n; i++)

{

if (a[i] != b[i])

{

cout << i + 1 << endl;

break;

}

}

}

## }

## CON4\_40 - Đếm Số 0

## #include <bits/stdc++.h>

## #define endl "\n"

## using namespace std;

## int main()

## {

## ios\_base::sync\_with\_stdio(0);

## cin.tie(0);

## int t;

## cin >> t;

## while (t--)

## {

## int n;

## cin >> n;

## int a[n];

## for (int i = 0; i < n; i++)

## cin >> a[i];

## int m = lower\_bound(a, a + n, 1) - a;

## cout << m << endl;

## }

## }

## CTDL061 - Xâu con chung dài nhất

## #include <bits/stdc++.h>

## using namespace std;

## int LCS(string s1, string s2)

## {

## int F[s1.size() + 1][s2.size() + 1] = {};

## for (int i = 0; i < s1.size(); i++)

## {

## for (int j = 0; j < s2.size(); j++)

## {

## if (s1[i] == s2[j])

## F[i + 1][j + 1] = F[i][j] + 1;

## else

## F[i + 1][j + 1] = max(F[i][j + 1], F[i + 1][j]);

## }

## }

## return F[s1.size()][s2.size()];

## }

## int main()

## {

## int t;

## cin >> t;

## while (t--)

## {

## string a, b;

## cin >> a >> b;

## cout << LCS(a, b) << endl;

## }

## }

## CTDL062 - Dãy con K phần tử có tổng bằng S

#include <bits/stdc++.h>

using namespace std;

int a[105], b[105],n, k, ok, m, cnt;

void Try(int pos1, int pos2, int s){

for(int i = pos1 + 1; i <= n; ++i){

if(s + a[i] == k){

ok=1;

b[pos2+1] = a[i];

if(pos2 + 2 == m) cnt++;

}

else if(s + a[i] < k){

b[pos2+1] = a[i];

Try(i, pos2+1, s + a[i]);

}

}

}

int main()

{

int t;

cin >>t;

while(t--){

ok=0, cnt=0;

cin >>n >> m>>k;

for(int i = 1; i <= n; ++i) cin >>a[i];

sort(a + 1, a + n + 1);

Try(0,-1,0);

cout <<cnt;

cout <<'\n';

}

return 0;

}

## CTDL063 - DÃY CON DÀI NHẤT CÓ TỔNG CHIA HẾT CHO K

#include <bits/stdc++.h>

using namespace std;

int main()

{

int t;

cin >> t;

while (t--)

{

int n, k;

cin >> n >> k;

vector<long long> a(n + 1);

vector<vector<long long>> f(n + 1, vector<long long>(k, 0));

for (int i = 1; i <= n; i++)

{

cin >> a[i];

a[i] %= k;

}

for (int i = 1; i < k; i++)

f[1][i] = -1e18;

f[1][a[1]] = 1;

for (int i = 2; i <= n; i++)

{

for (int j = 0; j < k; j++)

{

f[i][j] = max(f[i - 1][j], f[i - 1][(j - a[i] + k) % k] + 1);

}

}

cout << f[n][0] << endl;

}

}

## CON5\_5 - TỔ HỢP C(n, k)

#include <stdio.h>

int a[1001][1001], mod = 1e9 + 7;

int main()

{

a[0][0] = 1;

for (int i = 1; i < 1001; i++)

{

a[i][0] = 1;

for (int j = 1; j < 1001; j++)

a[i][j] = (a[i - 1][j - 1] % mod + a[i - 1][j] % mod) % mod;

}

int t;

scanf("%d", &t);

while (t--)

{

int n, k;

scanf("%d%d", &n, &k);

printf("%d\n", a[n][k]);

}

}

## CON5\_6 - XÂU CON ĐỐI XỨNG DÀI NHẤT

#include <bits/stdc++.h>

#define endl "\n"

using namespace std;

int main()

{

ios\_base::sync\_with\_stdio(0);

cin.tie(0);

int t;

cin >> t;

while (t--)

{

string s;

cin >> s;

int n = s.size(), ans = 1;

vector<vector<bool>> a(n, vector<bool>(n));

for (int i = 0; i < n; i++)

a[i][i] = 1;

for (int i = 1; i < n; i++)

for (int j = 0; j < n - i; j++)

{

int k = i + j;

a[j][k] = ((j + 1 > k - 1 || a[j + 1][k - 1]) && s[j] == s[k]);

if (a[j][k])

ans = i + 1;

}

cout << ans << endl;

}

}

## CON5\_7 - BẬC THANG

## #include <bits/stdc++.h>

## using namespace std;

## int main()

## {

## int t;

## cin >> t;

## while (t--)

## {

## int n, k, mod = 1e9 + 7;

## cin >> n >> k;

## long long a[n] = {}, b[n];

## for (int i = 0; i < min(n, k); i++)

## a[i] = 1;

## b[0] = 1;

## for (int i = 1; i < n; i++)

## {

## if (i <= k)

## a[i] = (a[i] + b[i - 1]) % mod;

## else

## {

## long long m = (b[i - 1] - b[i - k - 1]) % mod;

## if (m < 0)

## m += mod;

## a[i] = (a[i] + m) % mod;

## }

## b[i] = (b[i - 1] + a[i]) % mod;

## }

## cout << a[n - 1] << endl;

## }

## }

## CON5\_8 - HÌNH VUÔNG LỚN NHẤT

## #include <bits/stdc++.h>

## using namespace std;

## int main(){

## int t;

## cin >> t;

## while (t--)

## {

## int n, m, s = 0;

## cin >> n >> m;

## vector<vector<int>> a(n, vector<int>(m));

## for (int i = 0; i < n; i++)

## for (int j = 0; j < m; j++)

## cin >> a[i][j];

## for (int i = 0; i < m; i++)

## for (int j = 1; j < n; j++)

## if (a[j][i] == 1)

## a[j][i] = a[j - 1][i] + 1;

## for (int i = 0; i < n; i++)

## {

## stack<int> b, c;

## int l[m], r[m];

## for (int j = 0; j < m; j++)

## {

## while (!b.empty() && a[i][j] <= a[i][b.top()])

## b.pop();

## if (b.empty())

## l[j] = 0;

## else

## l[j] = b.top() + 1;

## b.push(j);

## }

## for (int j = m - 1; j >= 0; j--)

## {

## while (!c.empty() && a[i][j] <= a[i][c.top()])

## c.pop();

## if (c.empty())

## r[j] = m - 1;

## else

## r[j] = c.top() - 1;

## c.push(j);

## }

## for (int j = 0; j < m; j++)

## s = max(s, min(r[j] - l[j] + 1, a[i][j]));

## }

## cout << s << endl;

## }

## }

## CON5\_9 - SỐ CÓ TỔNG CHỮ SỐ BẰNG K

#include <bits/stdc++.h>

using namespace std;

int main()

{

long long f[105][50005] = {}, mod = 1e9 + 7;

for (int i = 1; i <= 9; i++)

f[1][i] = 1;

for (int i = 2; i <= 100; i++)

{

for (int j = 5000; j >= 0; j--)

{

for (int k = 0; k <= 9; k++)

{

if (j >= k)

{

f[i][j] += f[i - 1][j - k];

f[i][j] %= mod;

}

}

}

}

int t;

cin >> t;

while (t--)

{

int n, k;

cin >> n >> k;

cout << f[n][k] << endl;

}

## }

## CON5\_10 - ĐƯỜNG ĐI NHỎ NHẤT

## #include <bits/stdc++.h>

## #define endl "\n"

## using namespace std;

## int main()

## {

## int t;

## cin >> t;

## while (t--)

## {

## int n, m;

## cin >> n >> m;

## int a[n + 1][m + 1], b[n + 1][m + 1] = {};

## for (int i = 1; i <= n; i++)

## for (int j = 1; j <= m; j++)

## cin >> a[i][j];

## for (int i = 1; i <= n; i++)

## {

## for (int j = 1; j <= m; j++)

## {

## if (i == 1)

## b[i][j] = b[i][j - 1];

## else if (j == 1)

## b[i][j] = b[i - 1][j];

## else

## b[i][j] = min(b[i - 1][j - 1], min(b[i - 1][j], b[i][j - 1]));

## b[i][j] += a[i][j];

## }

## }

## cout << b[n][m] << endl;

## }

## }

## CON5\_26 - XEM PHIM

## #include <bits/stdc++.h>

## #define f(i,a,b) for(int i=a; i<=b; ++i)

## using namespace std;

## int dp[1005][1005];

## int main()

## {

## int t;

## cin >>t;

## while(t--){

## int s, n;

## cin >> s >> n;

## int a[n + 1];

## for (int i = 1; i <= n; i++)cin >> a[i];

## for (int i = 1; i <= n; i++)

## for (int j = 1; j <= s; j++)

## if (a[i] <= j) dp[i][j] = max(dp[i - 1][j - a[i]] + a[i], dp[i - 1][j]);

## else dp[i][j] = dp[i - 1][j];

## cout << dp[n][s] <<'\n';

## f(i,1,n) f(j,1,n) dp[i][j]=0;

## }

## return 0;

## }

## CTDL076 - Xâu nhị phân

## #include <bits/stdc++.h>

## const int MOD=1e9+7;

## using namespace std;

## string cvt1(long long n){

## string s="";

## while(n){

## if(n%2) s = "1" + s;

## else s = "0" + s;

## n/=2;

## }

## return s;

## }

## string addBinary(string binary1, string binary2) {

## while (binary1.length() < binary2.length()) {

## binary1 = "0" + binary1;

## }

## while (binary2.length() < binary1.length()) {

## binary2 = "0" + binary2;

## }

## int carry = 0;

## string result = "";

## for (int i = binary1.length() - 1; i >= 0; i--) {

## int sum = (binary1[i] - '0') + (binary2[i] - '0') + carry;

## if (sum == 0) {

## result = "0" + result;

## carry = 0;

## } else if (sum == 1) {

## result = "1" + result;

## carry = 0;

## } else if (sum == 2) {

## result = "0" + result;

## carry = 1;

## } else if (sum == 3) {

## result = "1" + result;

## carry = 1;

## }

## }

## return result;

## }

## int main()

## {

## int t;

## cin >> t;

## while(t--){

## int n, k;

## cin >>n >>k;

## string s;

## cin >>s;

## string tmp = cvt1(k+1);

## cout <<addBinary(tmp,s);

## cout <<endl;

## }

## return 0;

## }

## CTDL077 - Khoảng cách

## #include <bits/stdc++.h>

## #define ll long long

## #define all(v) v.begin(),v.end()

## #define f(i,a,b) for(int i=a; i<=b; ++i)

## using namespace std;

## ll tinh(ll n){

## if(n == 0) return 1;

## return n\*tinh(n-1);

## }

## int main()

## {

## int t;

## cin >> t;

## while (t--){

## int n;

## cin >>n;

## ll a[n+5], b[n+5], checka[n+5]={}, checkb[n+5]={}, posa=0, posb=0;

## f(i,1,n){

## int x = 0;

## cin >>a[i];

## checka[a[i]]++;

## for(int j = 1; j < a[i]; ++j) if(!checka[j]) x++;

## posa += x \* tinh(n-i);

## }

## f(i,1,n){

## int y = 0;

## cin >>b[i];

## checkb[b[i]]++;

## for(int j = 1; j < b[i]; ++j) if(!checkb[j]) y++;

## posb += y \* tinh(n-i);

## }

## cout <<abs(posa-posb) <<'\n';

## }

## return 0;

## }

## CTDL078 - Dãy chữ số

## #include <bits/stdc++.h>

## using namespace std;

## const int base = 1000000000; const int base\_digits = 9;

## struct bigint {

## vector<int> a; int sign;

## bigint() :

## sign(1) {

## }

## bigint(long long v) {

## \*this = v;

## }

## bigint(const string &s) {

## read(s);

## }

## void operator=(const bigint &v) {

## sign = v.sign;

## a = v.a;

## }

## void operator=(long long v) {

## sign = 1;

## if (v < 0)

## sign = -1, v = -v;

## for (; v > 0; v = v / base)

## a.push\_back(v % base);

## }

## bigint operator+(const bigint &v) const {

## if (sign == v.sign) {

## bigint res = v;

## for (int i = 0, carry = 0; i < (int) max(a.size(), v.a.size()) || carry; ++i) {

## if (i == (int) res.a.size())

## res.a.push\_back(0);

## res.a[i] += carry + (i < (int) a.size() ? a[i] : 0);

## carry = res.a[i] >= base;

## if (carry)

## res.a[i] -= base;

## }

## return res;

## }

## return \*this - (-v);

## }

## bigint operator-(const bigint &v) const {

## if (sign == v.sign) {

## if (abs() >= v.abs()) {

## bigint res = \*this;

## for (int i = 0, carry = 0; i < (int) v.a.size() || carry; ++i) {

## res.a[i] -= carry + (i < (int) v.a.size() ? v.a[i] : 0);

## carry = res.a[i] < 0;

## if (carry)

## res.a[i] += base;

## }

## res.trim();

## return res;

## }

## return -(v - \*this);

## }

## return \*this + (-v);

## }

## void operator\*=(int v) {

## if (v < 0)

## sign = -sign, v = -v;

## for (int i = 0, carry = 0; i < (int) a.size() || carry; ++i) {

## if (i == (int) a.size())

## a.push\_back(0);

## long long cur = a[i] \* (long long) v + carry;

## carry = (int) (cur / base);

## a[i] = (int) (cur % base);

## //asm("divl %%ecx" : "=a"(carry), "=d"(a[i]) : "A"(cur), "c"(base));

## }

## trim();

## }

## bigint operator\*(int v) const {

## bigint res = \*this;

## res \*= v;

## return res;

## }

## friend pair<bigint, bigint> divmod(const bigint &a1, const bigint &b1) {

## int norm = base / (b1.a.back() + 1);

## bigint a = a1.abs() \* norm;

## bigint b = b1.abs() \* norm;

## bigint q, r;

## q.a.resize(a.a.size());

## for (int i = a.a.size() - 1; i >= 0; i--) {

## r \*= base;

## r += a.a[i];

## int s1 = r.a.size() <= b.a.size() ? 0 : r.a[b.a.size()];

## int s2 = r.a.size() <= b.a.size() - 1 ? 0 : r.a[b.a.size() - 1];

## int d = ((long long) base \* s1 + s2) / b.a.back();

## r -= b \* d;

## while (r < 0)

## r += b, --d;

## q.a[i] = d;

## }

## q.sign = a1.sign \* b1.sign;

## r.sign = a1.sign;

## q.trim();

## r.trim();

## return make\_pair(q, r / norm);

## }

## bigint operator/(const bigint &v) const {

## return divmod(\*this, v).first;

## }

## bigint operator%(const bigint &v) const {

## return divmod(\*this, v).second;

## }

## void operator/=(int v) {

## if (v < 0)

## sign = -sign, v = -v;

## for (int i = (int) a.size() - 1, rem = 0; i >= 0; --i) {

## long long cur = a[i] + rem \* (long long) base;

## a[i] = (int) (cur / v);

## rem = (int) (cur % v);

## }

## trim();

## }

## bigint operator/(int v) const {

## bigint res = \*this;

## res /= v;

## return res;

## }

## int operator%(int v) const {

## if (v < 0)

## v = -v;

## int m = 0;

## for (int i = a.size() - 1; i >= 0; --i)

## m = (a[i] + m \* (long long) base) % v;

## return m \* sign;

## }

## void operator+=(const bigint &v) {

## \*this = \*this + v;

## }

## void operator-=(const bigint &v) {

## \*this = \*this - v;

## }

## void operator\*=(const bigint &v) {

## \*this = \*this \* v;

## }

## void operator/=(const bigint &v) {

## \*this = \*this / v;

## }

## bool operator<(const bigint &v) const {

## if (sign != v.sign)

## return sign < v.sign;

## if (a.size() != v.a.size())

## return a.size() \* sign < v.a.size() \* v.sign;

## for (int i = a.size() - 1; i >= 0; i--)

## if (a[i] != v.a[i])

## return a[i] \* sign < v.a[i] \* sign;

## return false;

## }

## bool operator>(const bigint &v) const {

## return v < \*this;

## }

## bool operator<=(const bigint &v) const {

## return !(v < \*this);

## }

## bool operator>=(const bigint &v) const {

## return !(\*this < v);

## }

## bool operator==(const bigint &v) const {

## return !(\*this < v) && !(v < \*this);

## }

## bool operator!=(const bigint &v) const {

## return \*this < v || v < \*this;

## }

## void trim() {

## while (!a.empty() && !a.back())

## a.pop\_back();

## if (a.empty())

## sign = 1;

## }

## bool isZero() const {

## return a.empty() || (a.size() == 1 && !a[0]);

## }

## bigint operator-() const {

## bigint res = \*this;

## res.sign = -sign;

## return res;

## }

## bigint abs() const {

## bigint res = \*this;

## res.sign \*= res.sign;

## return res;

## }

## long long longValue() const {

## long long res = 0;

## for (int i = a.size() - 1; i >= 0; i--)

## res = res \* base + a[i];

## return res \* sign;

## }

## friend bigint gcd(const bigint &a, const bigint &b) {

## return b.isZero() ? a : gcd(b, a % b);

## }

## friend bigint lcm(const bigint &a, const bigint &b) {

## return a / gcd(a, b) \* b;

## }

## void read(const string &s) {

## sign = 1;

## a.clear();

## int pos = 0;

## while (pos < (int) s.size() && (s[pos] == '-' || s[pos] == '+')) {

## if (s[pos] == '-')

## sign = -sign;

## ++pos;

## }

## for (int i = s.size() - 1; i >= pos; i -= base\_digits) {

## int x = 0;

## for (int j = max(pos, i - base\_digits + 1); j <= i; j++)

## x = x \* 10 + s[j] - '0';

## a.push\_back(x);

## }

## trim();

## }

## friend istream& operator>>(istream &stream, bigint &v) {

## string s;

## stream >> s;

## v.read(s);

## return stream;

## }

## friend ostream& operator<<(ostream &stream, const bigint &v) {

## if (v.sign == -1)

## stream << '-';

## stream << (v.a.empty() ? 0 : v.a.back());

## for (int i = (int) v.a.size() - 2; i >= 0; --i)

## stream << setw(base\_digits) << setfill('0') << v.a[i];

## return stream;

## }

## static vector<int> convert\_base(const vector<int> &a, int old\_digits, int new\_digits) {

## vector<long long> p(max(old\_digits, new\_digits) + 1);

## p[0] = 1;

## for (int i = 1; i < (int) p.size(); i++)

## p[i] = p[i - 1] \* 10;

## vector<int> res;

## long long cur = 0;

## int cur\_digits = 0;

## for (int i = 0; i < (int) a.size(); i++) {

## cur += a[i] \* p[cur\_digits];

## cur\_digits += old\_digits;

## while (cur\_digits >= new\_digits) {

## res.push\_back(int(cur % p[new\_digits]));

## cur /= p[new\_digits];

## cur\_digits -= new\_digits;

## }

## }

## res.push\_back((int) cur);

## while (!res.empty() && !res.back())

## res.pop\_back();

## return res;

## }

## typedef vector<long long> vll;

## static vll karatsubaMultiply(const vll &a, const vll &b) {

## int n = a.size();

## vll res(n + n);

## if (n <= 32) {

## for (int i = 0; i < n; i++)

## for (int j = 0; j < n; j++)

## res[i + j] += a[i] \* b[j];

## return res;

## }

## int k = n >> 1;

## vll a1(a.begin(), a.begin() + k);

## vll a2(a.begin() + k, a.end());

## vll b1(b.begin(), b.begin() + k);

## vll b2(b.begin() + k, b.end());

## vll a1b1 = karatsubaMultiply(a1, b1);

## vll a2b2 = karatsubaMultiply(a2, b2);

## for (int i = 0; i < k; i++)

## a2[i] += a1[i];

## for (int i = 0; i < k; i++)

## b2[i] += b1[i];

## vll r = karatsubaMultiply(a2, b2);

## for (int i = 0; i < (int) a1b1.size(); i++)

## r[i] -= a1b1[i];

## for (int i = 0; i < (int) a2b2.size(); i++)

## r[i] -= a2b2[i];

## for (int i = 0; i < (int) r.size(); i++)

## res[i + k] += r[i];

## for (int i = 0; i < (int) a1b1.size(); i++)

## res[i] += a1b1[i];

## for (int i = 0; i < (int) a2b2.size(); i++)

## res[i + n] += a2b2[i];

## return res;

## }

## bigint operator\*(const bigint &v) const {

## vector<int> a6 = convert\_base(this->a, base\_digits, 6);

## vector<int> b6 = convert\_base(v.a, base\_digits, 6);

## vll a(a6.begin(), a6.end());

## vll b(b6.begin(), b6.end());

## while (a.size() < b.size())

## a.push\_back(0);

## while (b.size() < a.size())

## b.push\_back(0);

## while (a.size() & (a.size() - 1))

## a.push\_back(0), b.push\_back(0);

## vll c = karatsubaMultiply(a, b);

## bigint res;

## res.sign = sign \* v.sign;

## for (int i = 0, carry = 0; i < (int) c.size(); i++) {

## long long cur = c[i] + carry;

## res.a.push\_back((int) (cur % 1000000));

## carry = (int) (cur / 1000000);

## }

## res.a = convert\_base(res.a, 6, base\_digits);

## res.trim();

## return res;

## }

## };

## int main() {

## int t;

## cin >> t;

## while(t--){

## int n, k, m;

## bigint ans, x;

## ans=1;

## x=1;

## cin >> n >> k >> m;

## for(int i=1; i <= k; ++i){

## ans \*= n;

## }

## for(int i = 1; i <=m; ++i) x\*=10;

## while(ans >= x){

## ans/=10;

## }

## cout <<ans <<endl;

## }

## return 0;

## }

## CTDL079 - Tìm hạng

#include <bits/stdc++.h>

using namespace std;

int a[105];

bool check(int n){

int x = sqrt(n);

return x\*x == n;

}

int main()

{

a[1] = 1, a[2] = 2;

for(int i = 3; i <= 55; ++i){

int k = i;

if(check(k)) a[k] = 1;

else{

a[k] = 1e9;

for(int j = 1; j <= k/2; ++j){

a[k] = min(a[j]+a[k-j],a[k]);

}

}

}

int t;

cin >> t;

while (t--){

int n;

cin >>n;

cout <<a[n] <<' ';

}

return 0;

}

## CTDL080 - Sinh xâu từ điển

#include <bits/stdc++.h>

using namespace std;

int main(){

int t;

cin >> t;

while (t--){

string s;

cin >>s;

int n = s.size();

for(int i = 0; i < n; ++i) if(s[i] >= 'A' && s[i] <= 'Z') s[i] += 'a' -'A';

int dp[404]={}, mx=-1e9;

for(int i = 0; i < n; ++i){

dp[s[i]] = max(dp[s[i]], 1);

for(int j = 'a'; j < s[i]; ++j){

dp[s[i]] = max(dp[s[i]], dp[j]+1);

}

mx = max(mx, dp[s[i]]);

}

cout <<mx <<'\n';

}

return 0;

}

## CTDL081\_Tree - Độ sâu

#include <iostream>

#include <string>

using namespace std;

int depth(string s, int& index) {

if (s[index] == 'l') {

return 0;

} else if (s[index] == 'n') {

index++;

int left\_depth = depth(s, index);

index++;

int right\_depth = depth(s, index);

return max(left\_depth, right\_depth) + 1;

} else {

return -1;

}

}

int main() {

int t;

cin >> t;

while (t--) {

int n;

cin >> n;

string s;

cin >> s;

int index = 0;

int d = depth(s, index);

cout << d << endl;

}

return 0;

}

## CTDL082 - Đảo ngược ngăn sếp

#include <bits/stdc++.h>

using namespace std;

int main(){

int t;

cin >>t;

while(t--){

int n, k;

cin >>n;

stack<int> st;

queue<int> q;

for(int i=1;i<=n;i++){

int x;

cin >>x;

st.push(x);

}

while(!st.empty()){

cout <<st.top() <<' ';

st.pop();

}

cout <<'\n';

}

return 0;

}

## CTDL084 - 0&9

## #include <bits/stdc++.h>

## using namespace std;

## int main()

## {

## int t;

## cin >> t;

## while (t--)

## {

## int n;

## cin >> n;

## queue<long long> q;

## q.push(9);

## while (q.size())

## {

## long long k = q.front();

## q.pop();

## if (k % n == 0)

## {

## cout << k << endl;

## break;

## }

## q.push(k \* 10);

## q.push(k \* 10 + 9);

## }

## }

## }

## CTDL085 - SỐ BDN 1

## #include <bits/stdc++.h>

## using namespace std;

## string n, k;

## bool check(string s)

## {

## return s.size() < n.size() || (s.size() == n.size() && s <= n);

## }

## int main()

## {

## int t;

## cin >> t;

## while (t--)

## {

## int d = 0;

## cin >> n;

## stack<string> st;

## st.push("1");

## while (st.size())

## {

## k = st.top();

## d++;

## st.pop();

## if (check(k + "0"))

## st.push(k + "0");

## if (check(k + "1"))

## st.push(k + "1");

## }

## cout << d << endl;

## }

## }

## CTDL087 - Biến đổi số

## #include <bits/stdc++.h>

## using namespace std;

## int main()

## {

## int t;

## cin >> t;

## while(t--){

## int s, t;

## cin >> s >> t;

## pair<int, int> u;

## u.first = s;

## u.second = 0;

## map<int, int> mp;

## queue<pair<int, int>> q;

## q.push(u);

## while (q.size()){

## u = q.front();q.pop();

## if (u.first == t){

## cout << u.second <<'\n';

## break;

## }

## if (u.first <= t && !mp[u.first \* 2]){

## mp[u.first \* 2]++;

## q.push({u.first \* 2, u.second + 1});

## }

## if (u.first > 1 && !mp[u.first - 1]){

## mp[u.first - 1]++;

## q.push({u.first - 1, u.second+ 1});

## }

## }

## }

## return 0;

## }

## CTDL088 - Hình Vuông

#include <bits/stdc++.h>

#define ll long long

#define f(i,a,b) for(int i=a; i<=b; ++i)

const int MOD=1e9+7;

using namespace std;

int s[15], a[15], z[15], ans=MOD, mx, b[15];

void Try(int k){

int num=0;

while(k){

int x=k&1, ok=1;

f(i,1,6) if(s[i]!=a[i]){ok=0; break;}

if(ok) {ans=min(ans,num); break;}

num++;

k>>=1;

f(i,1,6) z[i]=s[i];

if(!x){

s[1]=z[4], s[2]=z[1], s[4]=z[5], s[5]=z[2];

}

else{

s[2]=z[5], s[3]=z[2], s[5]=z[6], s[6]=z[3];

}

}

}

int main()

{

f(i,1,6) cin >> s[i], b[i]=s[i];

f(i,1,6) cin >> a[i];

mx=1<<20;

f(j,0,mx){

f(i,1,6) s[i]=b[i];

Try(j);

}

cout<<ans;

return 0;

}

## CTDL089 - GIEO HẠT

#include <bits/stdc++.h>

using namespace std;

int rx[] = {1, -1, 0, 0};

int ry[] = {0, 0, 1, -1};

int ans=0;

struct data

{

int x, y, s;

};

int a[505][505];

bool check( int m, int n)

{

for (int i = 1; i <= m; i++)

{

for (int j = 1; j <= n; j++)

{

if (a[i][j] == 1)

return false;

}

}

return true;

}

int main()

{

int t;

cin >>t;

while(t--){

int n, m;

cin >> n >> m;

struct data u, v;

queue<struct data> q;

u.s = 0;

for (int i = 1; i <= n; i++)

{

for (int j = 1; j <= m; j++)

{

cin >> a[i][j];

if (a[i][j] == 2)

{

u.x = i;

u.y = j;

q.push(u);

}

}

}

while (q.size())

{

u = q.front();

q.pop();

for (int i = 0; i < 4; i++)

{

v.x = u.x + rx[i];

v.y = u.y + ry[i];

if (v.x >= 1 && v.x <= n && v.y >= 1 && v.y <= m && a[v.x][v.y] == 1)

{

a[v.x][v.y] = 2;

v.s = u.s + 1;

ans = max(ans, v.s);

q.push(v);

}

}

}

if(check(n,m)) cout <<ans;

else cout <<-1;

cout << endl;

}

}

## CTDL095 - Biểu thức tương tương

#include <bits/stdc++.h>

using namespace std;

int main(){

int t;

cin >> t;

while (t--)

{

string s, k, ans = "";

cin >> s;

char c, d;

stack<char> st, ld;

for (int i = 0; i < s.size(); i++)

{

if (s[i] == '(')

{

st.push(s[i]);

ld.push(s[i - 1]);

}

else if (s[i] == ')')

{

k = "";

while (st.size())

{

c = st.top();

st.pop();

d = st.top();

st.pop();

if (ld.top() == '-')

{

k = c + k;

if (d == '-')

k = '+' + k;

else

k = '-' + k;

}

else

{

k = c + k;

k = d + k;

}

if (d == '(')

{

for (int j = 1; j < k.size(); j++)

st.push(k[j]);

break;

}

}

ld.pop();

}

else

st.push(s[i]);

}

while (st.size())

{

ans = st.top() + ans;

st.pop();

}

cout << ans << endl;

}

}

## CTDL096 - PREFIX TO INFIX

#include <bits/stdc++.h>

using namespace std;

string ts = "+-\*/";

bool check(char c)

{

for (int i = 0; i < ts.size(); i++)

if (ts[i] == c)

return 1;

return 0;

}

bool Var(char c)

{

return ((c >= 'a' && c <= 'z') || (c >= 'A' && c <= 'Z'));

}

int main(){

int t;

cin >> t;

while (t--)

{

string s, k;

cin >> s;

stack<string> st;

for (int i = s.size() - 1; i >= 0; i--)

{

if (check(s[i]))

{

k = '(' + st.top();

st.pop();

k += s[i];

k = k + st.top() + ')';

st.pop();

st.push(k);

}

else

st.push(string(1, s[i]));

}

cout << st.top() << endl;

}

}

## CTDL098 - ĐẢO NGƯỢC

#include<bits/stdc++.h>

using namespace std;

string reverseWord(string word){

stack<char> charStack;

for (char c : word)

{

charStack.push(c);

}

string reversedWord = "";

while (!charStack.empty())

{

reversedWord += charStack.top();

charStack.pop();

}

return reversedWord;

}

string reverseSentence(string sentence){

istringstream iss(sentence);

string word;

string reversedSentence = "";

while (iss >> word)

{

reversedSentence += reverseWord(word) + " ";

}

return reversedSentence;

}

int main(){

int T;

cin >> T;

for (int i = 0; i < T; ++i)

{

cin.ignore();

string sentence;

getline(cin, sentence);

cout << reverseSentence(sentence) << endl;

}

return 0;

}

## CTDL099 - PHẦN TỬ BÊN PHẢI ĐẦU TIÊN LỚN HƠN

## #include <bits/stdc++.h>

## using namespace std;

## int main(){

## int t;

## cin >> t;

## while (t--){

## int n;

## cin >> n;

## stack<int> st;

## vector<int> a(n), b(n, -1);

## for (int i = 0; i < n; i++)

## cin >> a[i];

## for (int i = 0; i < n; i++){

## while (!st.empty() && a[i] > a[st.top()]){

## b[st.top()] = i;

## st.pop();

## }

## st.push(i);

## }

## for (int i = 0; i < n; i++)

## {

## if (b[i] == -1)

## cout << -1 << " ";

## else

## cout << a[b[i]] << " ";

## }

## cout << endl;

## }

## }

## CTDL100 - HÌNH CHỮ NHẬT LỚN NHẤT

#include <bits/stdc++.h>

using namespace std;

int main(){

int t;

cin >> t;

while (t--){

int n;

cin >> n;

long long a[n] = {0}, MAX = 0;

for (int i = 0; i < n; i++)

cin >> a[i];

for (int i = 0; i < n; i++){

int pos1 = i, pos2 = i, s1 = 0, s2 = 0;

while (a[pos1] >= a[i] && pos1 >= 0){

pos1--;

s1++;

}

while (a[pos2] >= a[i] && pos2 < n){

pos2++;

s2++;

}

MAX = max(MAX, a[i] \* (s1 + s2 - 1));

}

cout << MAX << endl;

}

## }

## CTDL101 - CHUYỂN DANH SÁCH SANG DANH SÁCH KỀ

## #include <bits/stdc++.h>

## using namespace std;

## int main()

## {

## int t;

## cin >> t;

## while (t--)

## {

## int n, m, u, v;

## cin >> n >> m;

## vector<vector<int>> a(n + 1);

## while (m--)

## {

## cin >> u >> v;

## a[u].push\_back(v);

## a[v].push\_back(u);

## }

## for (int i = 1; i <= n; i++)

## {

## cout << i << ": ";

## for (int j = 0; j < a[i].size(); j++)

## {

## cout << a[i][j] << " ";

## }

## cout << endl;

## }

## }

## }

## CTDL102 - CHUYỂN MA TRẬN KỀ SANG DANH SÁCH KỀ

## #include <bits/stdc++.h>

## using namespace std;

## int main(){

## int n, x;

## cin >> n;

## vector<vector<int>> a(n + 1);

## for (int i = 1; i <= n; i++){

## for (int j = 1; j <= n; j++){

## cin >> x;

## if (x == 1){

## a[i].push\_back(j);

## }

## }

## }

## for (int i = 1; i <= n; i++){

## cout <<i <<": ";

## for (auto j : a[i])

## cout << j << " ";

## cout << endl;

## }

## return 0;

## }

## CTDL103 - CHUYỂN DANH SÁCH KỀ SANG MA TRẬN KỀ

## #include <bits/stdc++.h>

## using namespace std;

## int main(){

## int n;

## cin >> n;

## string s;

## vector<vector<int>> a(n + 1, vector<int>(n + 1, 0));

## cin.ignore();

## for (int i = 1; i <= n; i++){

## getline(cin, s);

## s += ' ';

## int k = 0;

## for (int j = 0; j < s.size(); j++){

## if (s[j] >= '0' && s[j] <= '9')

## k = k \* 10 + s[j] - '0';

## else{

## a[i][k] = 1;

## k = 0;

## }

## }

## }

## for (int i = 1; i <= n; i++){

## for (int j = 1; j <= n; j++)

## cout << a[i][j] << " ";

## cout << endl;

## }

## return 0;

## }

## CTDL104 - BIỂU DIỄN ĐỒ THỊ CÓ HƯỚNG

## #include <bits/stdc++.h>

## using namespace std;

## int main(){

## int t;

## cin >> t;

## while (t--){

## int n, m, u, v;

## cin >> n >> m;

## vector<vector<int>> a(n + 1);

## while (m--){

## cin >> u >> v;

## a[u].push\_back(v);

## }

## for (int i = 1; i <= n; i++){

## cout << i << ": ";

## for (int j = 0; j < a[i].size(); j++)

## cout << a[i][j] << " ";

## cout << endl;

## }

## }

## return 0;

## }

## CTDL105 - DFS TRÊN ĐỒ THỊ VÔ HƯỚNG

## #include <bits/stdc++.h>

## using namespace std;

## int main(){

## int t;

## cin >> t;

## while (t--){

## int n, m, u, x, y;

## cin >> n >> m >> u;

## vector<bool> check(n + 1, 1);

## vector<vector<int>> a(n + 1);

## stack<int> st;

## while (m--){

## cin >> x >> y;

## a[x].push\_back(y);

## a[y].push\_back(x);

## }

## for (int i = 1; i <= n; i++)

## sort(a[i].begin(), a[i].end());

## st.push(u);

## check[u] = 0;

## cout << u << " ";

## while (st.size()){

## u = st.top();

## st.pop();

## for (int i = 0; i < a[u].size(); i++){

## int v = a[u][i];

## if (check[v]){

## cout << v << " ";

## check[v] = 0;

## st.push(u);

## st.push(v);

## break;

## }

## }

## }

## cout << endl;

## }

## }

## CTDL106 - DFS TRÊN ĐỒ THỊ CÓ HƯỚNG

## #include <bits/stdc++.h>

## using namespace std;

## int main()

## {

## int t;

## cin >> t;

## while (t--)

## {

## int n, m, u, x, y;

## cin >> n >> m >> u;

## vector<bool> check(n + 1, 1);

## vector<vector<int>> a(n + 1);

## stack<int> st;

## while (m--)

## {

## cin >> x >> y;

## a[x].push\_back(y);

## }

## for (int i = 1; i <= n; i++)

## sort(a[i].begin(), a[i].end());

## st.push(u);

## check[u] = 0;

## cout << u << " ";

## while (st.size())

## {

## u = st.top();

## st.pop();

## for (int i = 0; i < a[u].size(); i++)

## {

## int v = a[u][i];

## if (check[v])

## {

## cout << v << " ";

## check[v] = 0;

## st.push(u);

## st.push(v);

## break;

## }

## }

## }

## cout << endl;

## }

## }

## CTDL107 - BFS TRÊN ĐỒ THỊ VÔ HƯỚNG

## #include <bits/stdc++.h>

## using namespace std;

## int main(){

## int t;

## cin >> t;

## while (t--)

## {

## int n, m, u, x, y;

## cin >> n >> m >> u;

## vector<int> a[n + 1], check(n + 1, 0);

## for (int i = 0; i < m; i++)

## {

## cin >> x >> y;

## a[x].push\_back(y);

## a[y].push\_back(x);

## }

## 

## queue<int> q;

## q.push(u);

## check[u] = 1;

## while (q.size())

## {

## u = q.front();

## q.pop();

## cout << u << " ";

## for (int i = 0; i < a[u].size(); i++)

## {

## int v = a[u][i];

## if (!check[v])

## {

## q.push(v);

## check[v] = 1;

## }

## }

## }

## cout << endl;

## }

## return 0;

## }

## CTDL108 - BFS TRÊN ĐỒ THỊ CÓ HƯỚNG

## #include <bits/stdc++.h>

## using namespace std;

## int main(){

## cin.tie(0);

## int t;

## cin >> t;

## while (t--)

## {

## int n, m, u, x, y;

## cin >> n >> m >> u;

## vector<int> a[n + 1], check(n + 1, 0);

## for (int i = 0; i < m; i++)

## {

## cin >> x >> y;

## a[x].push\_back(y);

## }

## for (int i = 1; i <= n; i++)

## sort(a[i].begin(), a[i].end());

## queue<int> q;

## q.push(u);

## check[u] = 1;

## while (q.size())

## {

## u = q.front();

## q.pop();

## cout << u << " ";

## for (int i = 0; i < a[u].size(); i++)

## {

## int v = a[u][i];

## if (!check[v])

## {

## q.push(v);

## check[v] = 1;

## }

## }

## }

## cout << endl;

## }

## }

## CTDL109 - TÌM ĐƯỜNG ĐI THEO DFS VỚI ĐỒ THỊ CÓ HƯỚNG

## #include <bits/stdc++.h>

## using namespace std;

## int main(){

## int t;

## cin >> t;

## while (t--)

## {

## int n, m, u, v, x, y, ok = 0;

## cin >> n >> m >> u >> v;

## string s;

## vector<bool> check(n + 1, 1);

## vector<vector<int>> a(n + 1);

## stack<pair<int, string>> st;

## while (m--)

## {

## cin >> x >> y;

## a[x].push\_back(y);

## }

## st.push({u, to\_string(u) + " "});

## check[u] = 0;

## while (st.size())

## {

## x = st.top().first;

## s = st.top().second;

## if (x == v)

## {

## ok = 1;

## cout << s << endl;

## break;

## }

## st.pop();

## for (int i = 0; i < a[x].size(); i++)

## {

## int y = a[x][i];

## if (check[y])

## {

## check[y] = 0;

## st.push({x, s});

## st.push({y, s + to\_string(y) + " "});

## break;

## }

## }

## }

## if (ok == 0)

## cout << -1 << endl;

## }

## return 0;

## }

## CTDL110 - ĐƯỜNG ĐI THEO BFS TRÊN ĐỒ THỊ VÔ HƯỚNG

## #include <bits/stdc++.h>

## using namespace std;

## int main(){

## int t;

## cin >> t;

## while (t--){

## int n, m, u, v, x, y, ok = 0;

## cin >> n >> m >> u >> v;

## string s;

## vector<bool> check(n + 1, 1);

## vector<vector<int>> a(n + 1);

## queue<pair<int, string>> st;

## while (m--)

## {

## cin >> x >> y;

## a[x].push\_back(y);

## a[y].push\_back(x);

## }

## st.push({u, to\_string(u) + " "});

## check[u] = 0;

## while (st.size())

## {

## x = st.front().first;

## s = st.front().second;

## if (x == v)

## {

## ok = 1;

## cout << s << endl;

## break;

## }

## st.pop();

## for (int i = 0; i < a[x].size(); i++)

## {

## int y = a[x][i];

## if (check[y])

## {

## check[y] = 0;

## st.push({y, s + to\_string(y) + " "});

## }

## }

## }

## if (ok == 0)

## cout << -1 << endl;

## }

## }

## CTDL111 - Bàn cờ 01

## #include<bits/stdc++.h>

## using namespace std;

## #define int long long

## const int maxn = 1e7 + 7;

## const int mod = 1e9 + 7;

## void implement(){

## int n ;

## cin >> n ;

## if(n == 1){

## cout << 1 ;

## return ;

## }

## cout << 2\*(n-1);

## }

## int32\_t main(){

## int t; cin >> t;

## while(t--){

## implement();

## cout << endl;

## }

## return 0;

## }

## SCTDLTree001 - Duyệt cây nhị phân 001

## #include <iostream>

## #include <queue>

## #include <vector>

## using namespace std;

## struct TreeNode {

## int val;

## TreeNode\* left;

## TreeNode\* right;

## TreeNode(int x) : val(x), left(NULL), right(NULL) {}

## };

## vector<vector<int>> levelOrder(TreeNode\* root) {

## vector<vector<int>> result;

## if (root == NULL) {

## return result;

## }

## queue<TreeNode\*> q;

## q.push(root);

## while (!q.empty()) {

## int levelSize = q.size();

## vector<int> levelValues;

## for (int i = 0; i < levelSize; i++) {

## TreeNode\* node = q.front();

## q.pop();

## levelValues.push\_back(node->val);

## if (node->left != NULL) {

## q.push(node->left);

## }

## if (node->right != NULL) {

## q.push(node->right);

## }

## }

## result.push\_back(levelValues);

## }

## return result;

## }

## int main() {

## int t;

## cin >> t;

## while (t--) {

## int n;

## cin >> n;

## vector<TreeNode\*> nodes(n);

## for (int i = 0; i < n; i++) {

## int val;

## cin >> val;

## if (val != -1) {

## nodes[i] = new TreeNode(val);

## }

## }

## for (int i = 0; i < n; i++) {

## if (nodes[i] != NULL) {

## int leftIndex = 2 \* i + 1;

## int rightIndex = 2 \* i + 2;

## if (leftIndex < n) {

## nodes[i]->left = nodes[leftIndex];

## }

## if (rightIndex < n) {

## nodes[i]->right = nodes[rightIndex];

## }

## }

## }

## vector<vector<int>> result = levelOrder(nodes[0]);

## for (const auto& level : result) {

## for (int val : level) {

## cout << val << " ";

## }

## cout << endl;

## }

## for (TreeNode\* node : nodes) {

## delete node;

## }

## }

## return 0;

## }

## SCTDLTree003 - Duyệt cây nhị phân 003

## #include <iostream>

## #include <queue>

## #include <vector>

## using namespace std;

## struct TreeNode {

## int val;

## TreeNode\* left;

## TreeNode\* right;

## TreeNode(int x) : val(x), left(NULL), right(NULL) {}

## };

## vector<vector<int>> levelOrderBottom(TreeNode\* root) {

## vector<vector<int>> result;

## if (root == NULL) {

## return result;

## }

## queue<TreeNode\*> q;

## q.push(root);

## while (!q.empty()) {

## int size = q.size();

## vector<int> level;

## for (int i = 0; i < size; i++) {

## TreeNode\* node = q.front();

## q.pop();

## level.push\_back(node->val);

## if (node->left != NULL) {

## q.push(node->left);

## }

## if (node->right != NULL) {

## q.push(node->right);

## }

## }

## result.insert(result.begin(), level);

## }

## return result;

## }

## int main() {

## int t;

## cin >> t;

## while (t--) {

## int n;

## cin >> n;

## vector<int> nums(n);

## for (int i = 0; i < n; i++) {

## cin >> nums[i];

## }

## TreeNode\* root = NULL;

## queue<TreeNode\*> q;

## if (nums[0] != -1) {

## root = new TreeNode(nums[0]);

## q.push(root);

## }

## int i = 1;

## while (!q.empty() && i < n) {

## TreeNode\* node = q.front();

## q.pop();

## if (nums[i] != -1) {

## node->left = new TreeNode(nums[i]);

## q.push(node->left);

## }

## i++;

## if (i < n && nums[i] != -1) {

## node->right = new TreeNode(nums[i]);

## q.push(node->right);

## }

## i++;

## }

## vector<vector<int>> result = levelOrderBottom(root);

## for (vector<int> level : result) {

## for (int num : level) {

## cout << num << " ";

## }

## cout << endl;

## }

## }

## return 0;

## }

## SCTDLTree004 - Tìm cây nhị phân 001

## #include <iostream>

## #include <vector>

## #include <queue>

## using namespace std;

## struct TreeNode {

## int val;

## TreeNode\* left;

## TreeNode\* right;

## TreeNode(int x) : val(x), left(NULL), right(NULL) {}

## };

## vector<int> rightSideView(TreeNode\* root) {

## vector<int> result;

## if (root == NULL) {

## return result;

## }

## queue<TreeNode\*> q;

## q.push(root);

## while (!q.empty()) {

## int size = q.size();

## for (int i = 0; i < size; i++) {

## TreeNode\* node = q.front();

## q.pop();

## 

## if (i == size - 1) {

## result.push\_back(node->val);

## }

## 

## if (node->left != NULL) {

## q.push(node->left);

## }

## 

## if (node->right != NULL) {

## q.push(node->right);

## }

## }

## }

## return result;

## }

## int main() {

## int t;

## cin >> t;

## 

## while (t--) {

## int n;

## cin >> n;

## 

## vector<int> nums(n);

## for (int i = 0; i < n; i++) {

## cin >> nums[i];

## }

## 

## TreeNode\* root = new TreeNode(nums[0]);

## queue<TreeNode\*> q;

## q.push(root);

## 

## int i = 1;

## while (!q.empty() && i < n) {

## TreeNode\* node = q.front();

## q.pop();

## 

## if (nums[i] != -1) {

## node->left = new TreeNode(nums[i]);

## q.push(node->left);

## }

## 

## i++;

## 

## if (i < n && nums[i] != -1) {

## node->right = new TreeNode(nums[i]);

## q.push(node->right);

## }

## 

## i++;

## }

## 

## vector<int> result = rightSideView(root);

## 

## for (int i = 0; i < result.size(); i++) {

## cout << result[i] << " ";

## }

## cout << endl;

## }

## return 0;

## }

## SCTDLTree006 - Độ sâu cây

## #include <iostream>

## #include <vector>

## #include <queue>

## using namespace std;

## struct TreeNode {

## int val;

## TreeNode\* left;

## TreeNode\* right;

## TreeNode(int x) : val(x), left(NULL), right(NULL) {}

## };

## int maxDepth(TreeNode\* root) {

## if (root == NULL) {

## return 0;

## }

## 

## queue<TreeNode\*> q;

## q.push(root);

## int depth = 0;

## 

## while (!q.empty()) {

## int size = q.size();

## 

## for (int i = 0; i < size; i++) {

## TreeNode\* node = q.front();

## q.pop();

## 

## if (node->left != NULL) {

## q.push(node->left);

## }

## 

## if (node->right != NULL) {

## q.push(node->right);

## }

## }

## 

## depth++;

## }

## 

## return depth;

## }

## int main() {

## int t;

## cin >> t;

## 

## while (t--) {

## int n;

## cin >> n;

## 

## vector<TreeNode\*> nodes(n);

## 

## for (int i = 0; i < n; i++) {

## int val;

## cin >> val;

## 

## if (val != -1) {

## nodes[i] = new TreeNode(val);

## }

## }

## 

## for (int i = 0; i < n; i++) {

## if (nodes[i] != NULL) {

## int leftIndex = 2 \* i + 1;

## int rightIndex = 2 \* i + 2;

## 

## if (leftIndex < n) {

## nodes[i]->left = nodes[leftIndex];

## }

## 

## if (rightIndex < n) {

## nodes[i]->right = nodes[rightIndex];

## }

## }

## }

## int result = maxDepth(nodes[0]);

## cout << result << endl;

## }

## return 0;

## }

## SCTDLTree007 - Độ sâu cây tối thiểu

#include <iostream>

#include <queue>

using namespace std;

struct Node {

int data;

Node\* left;

Node\* right;

};

Node\* createNode(int data) {

Node\* newNode = new Node();

if (!newNode) {

cout << "Memory error\n";

return NULL;

}

newNode->data = data;

newNode->left = newNode->right = NULL;

return newNode;

}

Node\* insertNode(Node\* root, int data) {

if (root == NULL) {

root = createNode(data);

return root;

}

queue<Node\*> q;

q.push(root);

while (!q.empty()) {

Node\* temp = q.front();

q.pop();

if (temp->left != NULL)

q.push(temp->left);

else {

temp->left = createNode(data);

return root;

}

if (temp->right != NULL)

q.push(temp->right);

else {

temp->right = createNode(data);

return root;

}

}

return root;

}

int findMinDepth(Node\* root) {

if (root == NULL)

return 0;

queue<Node\*> q;

q.push(root);

int depth = 0;

while (!q.empty()) {

int size = q.size();

depth++;

for (int i = 0; i < size; i++) {

Node\* temp = q.front();

q.pop();

if (temp->left == NULL && temp->right == NULL)

return depth;

if (temp->left != NULL)

q.push(temp->left);

if (temp->right != NULL)

q.push(temp->right);

}

}

return -1;

}

int main() {

int t;

cin >> t;

while (t--) {

int n;

cin >> n;

Node\* root = NULL;

for (int i = 0; i < n; i++) {

int data;

cin >> data;

root = insertNode(root, data);

}

int minDepth = findMinDepth(root);

cout << minDepth << endl;

}

return 0;

}

## SCTDLTree009 - Tổng đường đi 2

#include <iostream>

#include <vector>

struct TreeNode {

int val;

TreeNode\* left;

TreeNode\* right;

TreeNode(int x) : val(x), left(nullptr), right(nullptr) {}

};

void dfs(TreeNode\* node, int targetSum, std::vector<int>& path, std::vector<std::vector<int>>& result) {

if (node == nullptr) {

return;

}

path.push\_back(node->val);

if (node->left == nullptr && node->right == nullptr && node->val == targetSum) {

result.push\_back(path);

}

dfs(node->left, targetSum - node->val, path, result);

dfs(node->right, targetSum - node->val, path, result);

path.pop\_back();

}

std::vector<std::vector<int>> pathSum(TreeNode\* root, int targetSum) {

std::vector<std::vector<int>> result;

std::vector<int> path;

dfs(root, targetSum, path, result);

return result;

}

int main() {

// Test case input

int t;

std::cin >> t;

while (t--) {

int n;

std::cin >> n;

std::vector<TreeNode\*> nodes(n);

for (int i = 0; i < n; i++) {

int val;

std::cin >> val;

if (val != -1) {

nodes[i] = new TreeNode(val);

}

}

for (int i = 0; i < n; i++) {

if (nodes[i] == nullptr) {

continue;

}

int leftIndex = 2 \* i + 1;

int rightIndex = 2 \* i + 2;

if (leftIndex < n) {

nodes[i]->left = nodes[leftIndex];

}

if (rightIndex < n) {

nodes[i]->right = nodes[rightIndex];

}

}

int targetSum;

std::cin >> targetSum;

std::vector<std::vector<int>> result = pathSum(nodes[0], targetSum);

// Test case output

if (result.empty()) {

std::cout << "0" << std::endl;

} else {

for (const auto& path : result) {

std::cout << "[";

for (int i = 0; i < path.size(); i++) {

std::cout << path[i];

if (i != path.size() - 1) {

std::cout << ",";

}

}

std::cout << "]";

}

std::cout << std::endl;

}

// Clean up memory

for (int i = 0; i < n; i++) {

delete nodes[i];

}

}

return 0;

}

## SCTDLTree010 - Đường đi trên cây

#include <iostream>

#include <vector>

using namespace std;

struct TreeNode {

int val;

TreeNode\* left;

TreeNode\* right;

TreeNode(int x) : val(x), left(NULL), right(NULL) {}

};

void dfs(TreeNode\* node, vector<int>& path, vector<vector<int>>& paths) {

if (node == NULL) {

return;

}

path.push\_back(node->val);

if (node->left == NULL && node->right == NULL) {

paths.push\_back(path);

}

dfs(node->left, path, paths);

dfs(node->right, path, paths);

path.pop\_back();

}

vector<vector<int>> binaryTreePaths(TreeNode\* root) {

vector<vector<int>> paths;

vector<int> path;

dfs(root, path, paths);

return paths;

}

int main() {

int t;

cin >> t;

while (t--) {

int n;

cin >> n;

vector<int> tree(n);

for (int i = 0; i < n; i++) {

cin >> tree[i];

}

TreeNode\* root = NULL;

vector<TreeNode\*> nodes(n);

for (int i = 0; i < n; i++) {

if (tree[i] != -1) {

nodes[i] = new TreeNode(tree[i]);

if (i > 0) {

int parent = (i - 1) / 2;

if (i % 2 == 1) {

nodes[parent]->left = nodes[i];

} else {

nodes[parent]->right = nodes[i];

}

} else {

root = nodes[i];

}

}

}

vector<vector<int>> paths = binaryTreePaths(root);

for (const auto& path : paths) {

for (int i = 0; i < path.size(); i++) {

cout << path[i];

if (i < path.size() - 1) {

cout << "->";

}

}

cout << endl;

}

}

return 0;

}

## SCTDL075 - Hình thành mảng đích

#include <bits/stdc++.h>

using namespace std;

typedef long long ll;

const ll mod = 1e9 + 7;

int main() {

int t; cin >> t;

while (t--) {

int n; cin >> n;

vector<int> a(n);

for (int &x : a) cin >> x;

int res = a[0];

for (int i = 1; i < n; i ++) {

res += max(0, a[i] - a[i - 1]);

}

cout << res << endl;

}

return 0;

}

## CON4\_21 - Lũy thừa

#include <bits/stdc++.h>

using namespace std;

long long mod = 1e9 + 7;

long long poww(long long n, long long k)

{

if (k == 1)

return n;

long long x = poww(n, k / 2);

if (k % 2 == 0)

return (x \* x) % mod;

else

return (((x \* x) % mod) \* n) % mod;

}

int main(){

int t;

cin >> t;

while (t--)

{

long long n, k;

cin >> n >> k;

cout << poww(n, k) << endl;

}

}

## CON4\_23 - GẤP ĐÔI DÃY SỐ

#include <bits/stdc++.h>

using namespace std;

int main(){

int t;

cin >> t;

while (t--)

{

long long n, k;

cin >> n >> k;

long long a[n + 1];

for (long long i = 1; i <= n; i++)

{

a[i] = pow(2, i - 1);

}

for (int i = n; i >= 1; i--)

{

if (k == a[i])

{

cout << i << endl;

break;

}

else if (k > a[i])

k -= a[i];

}

}

}

## CON4\_25 - DÃY XÂU FIBONACI

#include <bits/stdc++.h>

using namespace std;

int main()

{

int t;

cin >> t;

while (t--)

{

long long n, k;

cin >> n >> k;

long long a[n + 1];

a[1] = 1;

a[2] = 1;

for (int i = 3; i <= n; i++)

a[i] = a[i - 1] + a[i - 2];

while (1)

{

if (n == 1)

{

cout << "A\n";

break;

}

else if (n == 2)

{

cout << "B\n";

break;

}

if (k > a[n - 2])

{

k -= a[n - 2];

n--;

}

else

n -= 2;

}

}

}

## CON4\_33 - Lũy Thừa Đảo

#include <bits/stdc++.h>

using namespace std;

long long mod = 1e9 + 7;

long long poww(long long n, long long k)

{

if (k == 0)

return 1;

long long x = poww(n, k / 2);

if (k % 2 == 0)

return (x \* x) % mod;

else

return (((x \* x) % mod) \* n) % mod;

}

int main()

{

int t;

cin >> t;

while (t--)

{

long long n, m, k;

cin >> n;

m = n;

k = 0;

while (m != 0)

{

k = k \* 10 + m % 10;

m /= 10;

}

cout << poww(n, k) << endl;

}

}

## CON7\_01 - Ngăn xếp 1

#include <bits/stdc++.h>

using namespace std;

int main(){

int n;

string s;

stack<int> st;

while (cin >> s)

{

if (s == "push")

{

cin >> n;

st.push(n);

}

else if (s == "pop")

st.pop();

else if (s == "show")

{

if (st.size())

{

stack<int> q;

while (st.size())

{

q.push(st.top());

st.pop();

}

while (q.size())

{

cout << q.top() << " ";

st.push(q.top());

q.pop();

}

cout << endl;

}

else

cout << "empty" << endl;

}

}

}

## CON7\_02 - Ngăn xếp 2

#include <bits/stdc++.h>

using namespace std;

int main(){

int t, n;

string s;

cin >> t;

stack<int> st;

while (t--)

{

cin >> s;

if (s == "PUSH")

{

cin >> n;

st.push(n);

}

else if (s == "POP")

{

if (st.size())

st.pop();

}

else if (s == "PRINT")

{

if (st.size())

cout << st.top() << endl;

else

cout << "NONE" << endl;

}

}

}

## CON7\_04 - Kiểm tra dãy ngoặc đúng

#include <bits/stdc++.h>

using namespace std;

int main(){

int t; cin >> t;

cin.ignore();

while(t--){

string str; getline(cin, str);

stack <char> stk;

bool ok = true;

for(int i = 0; i < str.size(); i ++){

if(str[i] == '{' || str[i] == '[' || str[i]=='(')

stk.push(str[i]);

else if(str[i]=='}'){

if(stk.empty()) ok = false;

else if(stk.top()=='{') stk.pop();

}

else if(str[i]==']'){

if(stk.empty()) ok = false;

else if(stk.top()=='[') stk.pop();

}

else if(str[i]==')'){

if(stk.empty()) ok = false;

else if(stk.top()=='(') stk.pop();

}

}

if(stk.size() > 0)

ok = false;

if(ok)

cout << "YES" << endl;

else

cout << "NO" << endl;

}

return 0;

}

## CON7\_05 - DÃY NGOẶC ĐÚNG DÀI NHẤT

#include <bits/stdc++.h>

using namespace std;

int main(){

int t;

cin >> t;

while (t--){

string s;

cin >> s;

stack<pair<char, int>> st;

int ans = 0, n = s.size();

vector<int> F(n, 0);

for (int i = 0; i < n; i++)

{

if (s[i] == ')' && !st.empty() && st.top().first == '(')

{

F[i] = F[i - 1];

if (st.top().second != 0)

F[i] += F[st.top().second - 1];

F[i] += 2;

st.pop();

}

else

st.push({s[i], i});

}

for (int i = 0; i < n; i++)

ans = max(ans, F[i]);

cout << ans << endl;

}

}

## CON7\_22 - HÌNH CHỮ NHẬT LỚN NHẤT

#include <bits/stdc++.h>

using namespace std;

int main(){

int t;

cin >> t;

while (t--)

{

int n;

cin >> n;

long long a[n] = {0}, MAX = 0;

for (int i = 0; i < n; i++)

cin >> a[i];

for (int i = 0; i < n; i++)

{

int pos1 = i, pos2 = i, s1 = 0, s2 = 0;

while (a[pos1] >= a[i] && pos1 >= 0)

{

pos1--;

s1++;

}

while (a[pos2] >= a[i] && pos2 < n)

{

pos2++;

s2++;

}

MAX = max(MAX, a[i] \* (s1 + s2 - 1));

}

cout << MAX << "\n";

}

}

## CON7\_21 - Phần tử bên phải nhỏ hơn

#include <bits/stdc++.h>

using namespace std;

int main(){

int t;

cin >> t;

while (t--){

int n;

cin >> n;

vector<int> a(n), l(n, -1), r(n, -1);

stack<int> b, c;

for (int i = 0; i < n; i++)

cin >> a[i];

for (int i = 0; i < n; i++)

{

while (!b.empty() && a[i] > a[b.top()])

{

l[b.top()] = i;

b.pop();

}

b.push(i);

}

for (int i = 0; i < n; i++)

{

while (!c.empty() && a[i] < a[c.top()])

{

r[c.top()] = i;

c.pop();

}

c.push(i);

}

for (int i = 0; i < n; i++)

{

if (l[i] == -1 || r[l[i]] == -1)

cout << -1 << " ";

else

cout << a[r[l[i]]] << " ";

}

cout << endl;

}

}

## CON7\_20 - PHẦN TỬ BÊN PHẢI ĐẦU TIÊN LỚN HƠN

#include <bits/stdc++.h>

using namespace std;

int main(){

int t;

cin >> t;

while (t--){

int n;

cin >> n;

stack<int> st;

vector<int> a(n), b(n, -1);

for (int i = 0; i < n; i++)

cin >> a[i];

for (int i = 0; i < n; i++){

while (!st.empty() && a[i] > a[st.top()])

{

b[st.top()] = i;

st.pop();

}

st.push(i);

}

for (int i = 0; i < n; i++){

if (b[i] == -1)

cout << -1 << " ";

else

cout << a[b[i]] << " ";

}

cout << endl;

}

}

## CON7\_19 - Biểu thức tăng giảm

#include <bits/stdc++.h>

using namespace std;

int main(){

int t;

cin >> t;

while (t--){

string s;

cin >> s;

stack<int> st;

int pos = 0;

for (int i = 0; i < s.size(); i++){

if (s[i] == 'D'){

if (i == 0 || s[i - 1] == 'I')

{

st.push(pos + 1);

pos++;

}

st.push(pos + 1);

pos++;

}

else{

while (st.size())

{

cout << st.top();

st.pop();

}

if (i == 0)

{

cout << pos + 1;

pos++;

}

if (s[i + 1] != 'D')

{

cout << pos + 1;

pos++;

}

}

}

while (st.size()){

cout << st.top();

st.pop();

}

cout << endl;

}

}

## CON7\_08 - Biểu thức tương đương

#include <bits/stdc++.h>

using namespace std;

int main(){

int t;

cin >> t;

while (t--)

{

string s, k, ans = "";

cin >> s;

char c, d;

stack<char> st, ld;

for (int i = 0; i < s.size(); i++)

{

if (s[i] == '(')

{

st.push(s[i]);

ld.push(s[i - 1]);

}

else if (s[i] == ')')

{

k = "";

while (st.size())

{

c = st.top();

st.pop();

d = st.top();

st.pop();

if (ld.top() == '-')

{

k = c + k;

if (d == '-')

k = '+' + k;

else

k = '-' + k;

}

else

{

k = c + k;

k = d + k;

}

if (d == '(')

{

for (int j = 1; j < k.size(); j++)

st.push(k[j]);

break;

}

}

ld.pop();

}

else

st.push(s[i]);

}

while (st.size())

{

ans = st.top() + ans;

st.pop();

}

cout << ans << endl;

}

}

## CON7\_09 - Xóa dầu ngoặc

#include<bits/stdc++.h>

using namespace std;

string s;

vector<pair<int,int>> pos;

set<string> res;

void Try(int i = 0, string k = ""){

    if(i == pos.size()){

        if(k == string(pos.size(),'0')) return;

        unordered\_set<int> st;

        for(int j = 0; j < pos.size(); ++j){

            if(k[j] == '1'){

                st.insert(pos[j].first);

                st.insert(pos[j].second);

            }

        }

        string tmp = "";

        for(int j = 0; j < s.length(); ++j){

            if(st.find(j) == st.end()){

            tmp += s[j];

            }

        }

        res.insert(tmp);

        return;

    }

    Try(i + 1, k + "0");

    Try(i + 1, k + "1");

}

int main(){

    cin >> s;

    stack<int> st;

    for(int i = 0; i < s.length(); ++i){

        if(s[i] == '(')

            st.push(i);

        if(s[i] == ')'){

            pos.push\_back({st.top(), i});

            st.pop();

        }

    }

    Try();

        for(string i: res){

            cout << i << '\n';

        }

    return 0;

}

## CON4\_22 - Tìm kiếm nhị phân

#include <bits/stdc++.h>

using namespace std;

int main(){

int t;

cin >> t;

while (t--){

int n, k;

cin >> n >> k;

int a[n];

for (int i = 0; i < n; i++){

cin >> a[i];

}

int m = lower\_bound(a, a + n, k) - a;

if (a[m] != k)

cout << "NO" << endl;

else

cout << m + 1 << endl;

}

}

## CON4\_26 - Hệ số K

#include<bits/stdc++.h>

using namespace std;

string add(string a,string b,int k){

while(a.size()<b.size())a="0"+a;

while(b.size()<a.size())b="0"+b;

int nho=0,l=a.size();

string ans="";

for(int i=l-1;i>=0;i--){

int so=a[i]-'0'+b[i]-'0'+nho;

nho=so/k;

ans=(char)(so%k+'0')+ans;

}

if(nho>0)ans=(char)(nho+'0')+ans;

return ans;

}

int main(){

int t;

cin >> t;

while (t--)

{

string a,b;

int k;

cin>>k>>a>>b;

cout<<add(a,b,k)<<"\n";

}

return 0;

}

## CON4\_28 - SẮP XẾP KANGURU

#include <bits/stdc++.h>

using namespace std;

int main(){

int t;

cin >> t;

while (t--){

int n;

cin >> n;

vector<int> a(n);

for (int i = 0; i < n; i++)

cin >> a[i];

sort(a.begin(), a.end());

int ans = n, l = n / 2 - 1, r = n - 1;

while (l >= 0 && r >= n / 2){

if (2 \* a[l] <= a[r])

{

l--;

r--;

ans--;

}

else

l--;

}

cout << ans << endl;

}

}

## CON4\_32 - CẶP NGHỊCH THẾ

#include <bits/stdc++.h>

using namespace std;

long long Merge(long long \*a, long long l, long long r, long long mid){

long long p1 = mid - l + 1, p2 = r - mid, ans = 0;

long long L[p1], R[p2];

for (int i = 0; i < p1; i++)

L[i] = a[l + i];

for (int i = 0; i < p2; i++)

R[i] = a[mid + i + 1];

int i = 0, j = 0, k = l;

while (i < p1 and j < p2){

if (L[i] <= R[j])

a[k++] = L[i++];

else

{

a[k++] = R[j++];

ans += p1 - i;

}

}

while (i < p1)

a[k++] = L[i++];

while (j < p2)

a[k++] = R[j++];

return ans;

}

long long calc(long long \*a, int l, int r){

long long ans = 0;

int mid;

if (l < r){

mid = (l + r) >> 1;

ans += calc(a, l, mid);

ans += calc(a, mid + 1, r);

ans += Merge(a, l, r, mid);

}

return ans;

}

int main(){

int t;

cin >> t;

while (t--){

int n;

cin >> n;

long long a[n];

for (int i = 0; i < n; i++)

cin >> a[i];

cout << calc(a, 0, n - 1) << endl;

}

}

## CON4\_34 - TÍCH ĐA THỨC

#include <stdio.h>

int main(){

int t;

scanf("%d", &t);

while (t--){

int n, m;

scanf("%d%d", &n, &m);

int a[n + 1], b[m + 1], c[m + n + 1] = {};

for (int i = n; i >= 1; i--)

scanf("%d", &a[i]);

for (int i = m; i >= 1; i--)

scanf("%d", &b[i]);

for (int i = 1; i <= n; i++)

{

for (int j = 1; j <= m; j++)

{

c[i + j] += a[i] \* b[j];

}

}

for (int i = m + n; i >= 2; i--)

printf("%d ", c[i]);

printf("\n");

}

}

## CON4\_36 - Tích hai số nhị phân

#include <bits/stdc++.h>

using namespace std;

int main(){

int t;

cin >> t;

while (t--){

string a, b;

cin >> a >> b;

long long s1 = 0, s2 = 0;

long long n1 = a.size(), n2 = b.size();

for (long long i = n1 - 1; i >= 0; i--)

s1 += (a[i] - '0') \* pow(2, n1 - i - 1);

for (long long i = n2 - 1; i >= 0; i--)

s2 += (b[i] - '0') \* pow(2, n2 - i - 1);

cout << s1 \* s2 << endl;

}

}

## CON4\_39 - PHẦN TỬ KHÁC NHAU

#include <bits/stdc++.h>

using namespace std;

int main(){

int t;

cin >> t;

while (t--){

int n;

cin >> n;

int a[n], b[n];

for (int i = 0; i < n; i++)

cin >> a[i];

for (int i = 0; i < n - 1; i++)

cin >> b[i];

for (int i = 0; i < n; i++){

if (a[i] != b[i]){

cout << i + 1 << endl;

break;

}

}

}

}

## CON4\_29 - Cặp điểm gần nhất

#include <bits/stdc++.h>

using namespace std;

double distance(pair<double, double> a, pair<double, double> b){

return sqrt((a.first - b.first) \* (a.first - b.first) + (a.second - b.second) \* (a.second - b.second));

}

int main(){

int t;

cin >> t;

while (t--)

{

int n;

cin >> n;

double ans = 1e9;

vector<pair<double, double>> a(n), b(n);

for (int i = 0; i < n; i++)

{

cin >> a[i].first >> a[i].second;

b[i].second = a[i].first;

b[i].first = a[i].second;

}

sort(a.begin(), a.end());

sort(b.begin(), b.end());

for (int i = 1; i < n; i++)

{

ans = min(ans, distance(a[i], a[i - 1]));

ans = min(ans, distance(b[i], b[i - 1]));

}

cout << setprecision(6) << fixed << ans << endl;

}

}

## CON4\_31 - LŨY THỪA MA TRẬN

#include <bits/stdc++.h>

using namespace std;

struct data{

long long x[10][10];

};

int n;

long long mod = 1000000007;

struct data operator\*(struct data a, struct data b){

struct data xx;

for (int i = 0; i < n; i++){

for (int j = 0; j < n; j++){

xx.x[i][j] = 0;

for (int k = 0; k < n; k++)

xx.x[i][j] = (xx.x[i][j] + (a.x[i][k] \* b.x[k][j]) % mod) % mod;

}

}

return xx;

};

struct data poww(struct data a, long long k){

if (k == 1)

return a;

if (k % 2 == 1)

return a \* poww(a, k - 1);

struct data xx = poww(a, k / 2);

return xx \* xx;

}

int main(){

int t;

cin >> t;

while (t--){

long long k;

cin >> n >> k;

struct data a;

for (int i = 0; i < n; i++)

for (int j = 0; j < n; j++)

cin >> a.x[i][j];

a = poww(a, k);

for (int i = 0; i < n; i++){

for (int j = 0; j < n; j++)

cout << a.x[i][j] << " ";

cout << endl;

}

}

}

## CON7\_03 - Đảo từ

#include<bits/stdc++.h>

using namespace std;

int main(){

int t;

cin >> t;

cin.ignore();

while (t--){

vector<string>ans;

string s;

getline(cin,s);

stack<char>st;

int j;

for(int i=0;i<s.size();i++){

if(s[i]!=' '){

j=i;

while(j<s.size()&&s[j]!=' '){

st.push(s[j]);

j++;

}

string tmp="";

while(!st.empty()){

tmp+=st.top();

st.pop();

}

ans.push\_back(tmp);

i=j-1;

}

}

for(string x:ans){

cout<<x<<" ";

}

cout<<"\n";

}

return 0;

}

## CON7\_06 - KIỂM TRA BIỂU THỨC SỐ HỌC

#include <bits/stdc++.h>

using namespace std;

int main(){

int t;

cin >> t;

cin.ignore();

while (t--){

int ok = 0;

string s;

getline(cin, s);

stack<char> st;

for (int i = 0; i < s.size(); i++){

if (s[i] == '(' || s[i] == '+' || s[i] == '-' || s[i] == '\*' || s[i] == '/')

st.push(s[i]);

if (s[i] == ')'){

if (st.top() == '(')

{

ok = 1;

break;

}

while (st.size() && st.top() != '(')

st.pop();

st.pop();

}

}

if (ok)

cout << "Yes" << endl;

else

cout << "No" << endl;

}

}

## CON6\_2 - SẮP XẾP THEO GIÁ TRỊ TUYỆT ĐỐI

#include<bits/stdc++.h>

using namespace std;

long long x;

void nhap(vector<long long> &a){

vector<long long>:: iterator it;

for(it = a.begin();it != a.end();it++){

cin >> \*it;

}

}

void in(vector<long long> a){

vector<long long>:: iterator it;

for(it = a.begin();it != a.end();it++){

cout << \*it << " ";

}

}

bool compare(long long a,long long b){

return abs(x - a) < abs(x - b);

}

void solution(){

int t;

cin >> t;

while(t--){

long long n;

cin >> n >> x;

vector<long long> a(n);

nhap(a);

stable\_sort(a.begin(),a.end(),compare);

in(a);

cout << endl;

}

}

int main(){

solution();

return 0;

}

## CON6\_4 - HỢP VÀ GIAO CỦA HAI DÃY SỐ 1

#include <bits/stdc++.h>

using namespace std;

int main(){

int t; cin >> t;

while(t--){

int n, m; cin >> n >> m;

vector<int> a(n);

set<int> st, sm;

for(int i = 0; i < n; i ++){

cin >> a[i];

st.insert(a[i]);

}

int k = 0;

for(int i = 1; i <= m; i ++){

int x; cin >> x;

st.insert(x);

if(n + i - st.size() - k != 0){

sm.insert(x);

k++;

}

}

for(auto x: st) cout << x << " ";

cout << endl;

for(auto x: sm) cout << x << " ";

cout << endl;

}

return 0;

}

## CON6\_3 - Đổi chỗ ít nhất

#include <bits/stdc++.h>

using namespace std;

int main(){

int t;

cin >> t;

while (t--)

{

int n, k, ans = 0;

cin >> n;

int a[n];

for (int i = 0; i < n; i++)

cin >> a[i];

for (int i = 0; i < n - 1; i++)

{

k = i;

for (int j = i + 1; j < n; j++)

if (a[j] < a[k])

k = j;

if (k != i)

ans++;

swap(a[k], a[i]);

}

cout << ans << endl;

}

}

## CON6\_7 - Sắp xếp dãy con liên tục

#include <bits/stdc++.h>

using namespace std;

int main(){

    int t;

    cin >> t;

    while (t--){

        int n;

        cin >> n;

        int a[n + 1], b[n + 1], s = 0, l, r, ma = 0, mi = 1e9;

        for (int i = 1; i <= n; i++){

            cin >> a[i];

            if (a[i] > s)

                s = a[i];

            b[i] = s;

        }

        for (int i = 1; i <= n - 1; i++){

            if (a[i] > a[i + 1]){

                l = i;

                break;

            }

        }

        for (int i = n; i >= 2; i--){

            if (a[i] < a[i - 1]){

                r = i;

                break;

            }

        }

        for (int i = l; i <= r; i++){

            if (a[i] > ma)

                ma = a[i];

            if (a[i] < mi)

                mi = a[i];

        }

        for (int i = 1; i <= n; i++){

            if (a[i] > mi){

                cout << i << " ";

                break;

            }

        }

        for (int i = n; i >= 1; i--){

            if (a[i] <= ma){

                cout << i << endl;

                break;

            }

        }

    }

}

## CON6\_8 - Đếm cặp

#include <iostream>

#include <vector>

#include <algorithm>

using namespace std;

int countPairs(vector<int>& X, vector<int>& Y) {

int count = 0;

int n = X.size();

int m = Y.size();

sort(X.begin(), X.end());

sort(Y.begin(), Y.end());

for (int i = 0; i < n; i++) {

if (X[i] == 0)

continue;

if (X[i] == 1) {

count += 0;

} else {

int idx = upper\_bound(Y.begin(), Y.end(), X[i]) - Y.begin();

count += m - idx;

if (X[i] == 2)

count--;

}

}

return count;

}

int main() {

int T;

cin >> T;

while (T--) {

int n, m;

cin >> n >> m;

vector<int> X(n);

vector<int> Y(m);

for (int i = 0; i < n; i++) {

cin >> X[i];

}

for (int i = 0; i < m; i++) {

cin >> Y[i];

}

int result = countPairs(X, Y);

cout << result << endl;

}

return 0;

}

## CON6\_9 - Số cặp tổng bằng K

#include <bits/stdc++.h>

using namespace std;

int main(){

int t;

cin >> t;

while (t--){

int n, k, x, s = 0;

cin >> n >> k;

int a[n];

map<int, int> m;

for (int i = 0; i < n; i++){

cin >> x;

m[x]++;

}

for (auto i : m){

if (i.first \* 2 == k)

s += i.second \* (i.second - 1);

else

s += i.second \* m[k - i.first];

}

s /= 2;

cout << s << endl;

}

}

## CON6\_10 - Sắp xếp chữ số

#include <bits/stdc++.h>

using namespace std;

int main(){

int t;

cin >> t;

while (t--){

int n, x, a[10] = {};

cin >> n;

for (int i = 0; i < n; i++){

cin >> x;

while (x != 0){

int k = x % 10;

a[k]++;

x /= 10;

}

}

for (int i = 0; i < 10; i++){

if (a[i] > 0)

cout << i << " ";

}

cout << endl;

}

}

## CON6\_11 - Tổng gần nhất 0

#include <bits/stdc++.h>

using namespace std;

int main(){

int t;

cin >> t;

while (t--){

int n, s = 1e7;

cin >> n;

int a[n];

for (int i = 0; i < n; i++)

cin >> a[i];

for (int i = 0; i < n - 1; i++){

for (int j = i + 1; j < n; j++){

if (abs(a[i] + a[j]) < abs(s))

s = a[i] + a[j];

}

}

cout << s << endl;

}

}

## CON6\_12 - Tìm k phần tử max đầu tiên

#include<bits/stdc++.h>

using namespace std;

int main(){

int t;

cin >> t;

while (t--){

int n,k;

cin>>n>>k;

int a[n];

for(int &x:a)cin>>x;

sort(a,a+n,greater<int>());

for(int i=0;i<k;i++){

cout<<a[i]<<" ";

}

cout<<"\n";

}

return 0;

}

## CON6\_13 - Số lần xuất hiện

#include <bits/stdc++.h>

using namespace std;

int main(){

int t;

cin >> t;

while (t--){

int n, k, s = 0;

cin >> n >> k;

int a[n];

for (int i = 0; i < n; i++){

cin >> a[i];

if (a[i] == k)

s++;

}

if (s == 0)

cout << -1 << endl;

else

cout << s << endl;

}

}

## CON6\_14 - Tổng Cặp số nguyên tố

#include <bits/stdc++.h>

using namespace std;

bool nto(int n){

if (n < 2)

return 0;

for (int i = 2; i <= sqrt(n); i++){

if (n % i == 0)

return 0;

}

return 1;

}

int main(){

ios\_base::sync\_with\_stdio(0);

cin.tie(0);

int t;

cin >> t;

while (t--){

int n, ok = 0;

cin >> n;

for (int i = 2; i <= n / 2; i++){

if (nto(i) == 1 && nto(n - i) == 1){

ok = 1;

cout << i << " " << n - i << endl;

break;

}

}

if (ok == 0)

cout << -1 << endl;

}

}

## CON6\_15 - MERGE SORT

#include <bits/stdc++.h>

using namespace std;

int main(){

ios\_base::sync\_with\_stdio(0);

cin.tie(0);

int t;

cin >> t;

while (t--){

int n;

cin >> n;

vector<int> a(n);

for (int i = 0; i < n; i++)

cin >> a[i];

sort(a.begin(), a.end());

for (int i = 0; i < n; i++)

cout << a[i] << " ";

cout << endl;

}

}

## CON6\_16 - QUICK SORT

#include<iostream>

#include<bits/stdc++.h>

using namespace std;

void Quicksort(){

int n;

cin>>n;

int arr[n];

for(int i=0;i<n;i++){

cin>>arr[i];

}

sort(arr,arr+n);

for(int i=0;i<n;i++){

cout<<arr[i]<<" ";

}

cout<<endl;

}

int main(){

int test;

cin>>test;

while(test--){

Quicksort();

}

}

## CON6\_17 - tích giữa phần tử lớn nhất và nhỏ nhất

#include <iostream>

#include <vector>

#include <algorithm>

int main() {

int T;

std::cin >> T;

while (T--) {

int n, m;

std::cin >> n >> m;

std::vector<int> A(n);

std::vector<int> B(m);

for (int i = 0; i < n; i++) {

std::cin >> A[i];

}

for (int i = 0; i < m; i++) {

std::cin >> B[i];

}

int maxA = \*std::max\_element(A.begin(), A.end());

int minB = \*std::min\_element(B.begin(), B.end());

int result = maxA \* minB;

std::cout << result << std::endl;

}

return 0;

}

## CON6\_18 - Hợp nhất dãy số

#include<bits/stdc++.h>

using namespace std;

int main(){

int t;

cin >> t;

while (t--){

int n,m;

cin>>n>>m;

vector<int>v;

for(int i=0;i<n+m;i++){

int x;cin>>x;

v.push\_back(x);

}

sort(v.begin(),v.end());

for(int &x:v)cout<<x<<" ";

cout<<"\n";

}

return 0;

}

## CON6\_19 - Bổ sung phần tử

#include <bits/stdc++.h>

using namespace std;

int main(){

int t;

cin >> t;

while (t--){

int n, s = 0;

cin >> n;

int a[n];

for (int i = 0; i < n; i++)

cin >> a[i];

sort(a, a + n);

for (int i = 1; i < n; i++)

if (a[i] != a[i - 1])

s += a[i] - a[i - 1] - 1;

cout << s << endl;

}

}

## CON6\_20 - Đếm cặp theo điều kiện

#include<bits/stdc++.h>

using namespace std;

int main(){

int t;

cin >> t;

while (t--){

int n,k;

cin>>n>>k;

int a[n];

for(int &x:a)cin>>x;

int cnt=0;

for(int i=0;i<n-1;i++){

for(int j=i+1;j<n;j++){

if(abs(a[j]-a[i])<k){

cnt++;

}

}

}

cout<<cnt<<"\n";

}

return 0;

}

## CON6\_21 - Số lần xuất hiện

#include<bits/stdc++.h>

using namespace std;

bool cmp(pair<int,int>a,pair<int,int>b){

if(a.second==b.second) return a.first<b.first;

return a.second>b.second;

}

int main(){

int t;

cin >> t;

while (t--)

{

int n;

cin>>n;

int a[n];

map<int,int>mp;

for(int i=0;i<n;i++){

cin>>a[i];

mp[a[i]]++;

}

int l=mp.size();

pair<int,int>p[l];

int i=0;

for(auto it:mp){

p[i++]=make\_pair(it.first,it.second);

}

sort(p,p+l,cmp);

for(int i=0;i<l;i++){

for(int j=0;j<p[i].second;j++){

cout<<p[i].first<<" ";

}

}

cout<<"\n";

}

return 0;

}

## CON6\_22 - Tìm kiếm trong mảng

#include <iostream>

using namespace std;

int binarySearch(int arr[], int n, int x) {

int left = 0, right = n - 1;

while (left <= right) {

int mid = left + (right - left) / 2;

if (arr[mid] == x) {

return 1;

}

if (arr[mid] < x) {

left = mid + 1;

} else {

right = mid - 1;

}

}

return -1;

}

int main() {

int T;

cin >> T;

while (T--) {

int n, x;

cin >> n >> x;

int arr[n];

for (int i = 0; i < n; i++) {

cin >> arr[i];

}

int result = binarySearch(arr, n, x);

cout << result << endl;

}

return 0;

}

## CON6\_24 - Tìm kiếm vòng

#include<bits/stdc++.h>

using namespace std;

void nhap(vector<long long> &a){

vector<long long>:: iterator it;

for(it = a.begin();it != a.end();it++){

cin >> \*it;

}

}

long long search(vector<long long> a,long long k){

int index = 0;

for(int i = 0;i < a.size();i++){

if(a[i] == k){

index = i;

break;

}

}

return index + 1;

}

void solution(){

int t;

cin >> t;

while(t--){

long long n,k;

cin >> n >> k;

vector<long long> a(n);

nhap(a);

cout << search(a,k) << endl;

}

}

int main(){

solution();

return 0;

}

## CON6\_27 - Sắp xếp chèn

#include <bits/stdc++.h>

using namespace std;

int main(){

int t;

cin>>t;

while(t--){

int n;

cin >> n;

int a[n];

for (int i = 0; i < n; i++)

cin >> a[i];

for (int i = 0; i < n; i++){

int k = a[i];

int j = i - 1;

while (j >= 0 && a[j] > k)

{

a[j + 1] = a[j];

j = j - 1;

}

a[j + 1] = k;

cout << "Buoc " << i << ": ";

for (j = 0; j <= i; j++)

cout << a[j] << " ";

cout << endl;

}

}

}

## CON7\_10 - SO SÁNH BIỂU THỨC

#include<bits/stdc++.h>

using namespace std;

string tim\_bieu\_thuc\_tuong\_duong(string s){

stack<int> st;

string res;

for(int i = 0;i < s.size();i++){

if(s[i] == '('){

st.push(i);

}

else if(s[i] == ')'){

if(st.size() > 0){

int tmp = st.top();st.pop();

if(s[tmp - 1] == '-'){

for(int j = tmp;j <= i;j++){

if(s[j] == '-'){

s[j] = '+';

}

else if(s[j] == '+'){

s[j] = '-';

}

}

}

}

}

}

for(int i = 0;i < s.size();i++){

if(s[i] != '(' && s[i] != ')'){

res.push\_back(s[i]);

}

}

return res;

}

bool ktra(string a,string b){

if(a == b){

return true;

}

return false;

}

void solution(){

int t;

cin >> t;

while(t--){

string a,b;

cin >> a >> b;

a = tim\_bieu\_thuc\_tuong\_duong(a);

b = tim\_bieu\_thuc\_tuong\_duong(b);

if(ktra(a,b) == true){

cout << "YES" << endl;

}

else{

cout << "NO" << endl;

}

}

}

int main(){

solution();

return 0;

}

## CON6\_29 - SẮP XẾP NỔI BỌT

#include <bits/stdc++.h>

using namespace std;

int main(){

int t;

cin>>t;

while(t--){

int n, b = 1;

cin >> n;

int a[n];

for (int i = 0; i < n; i++)

cin >> a[i];

for (int i = 0; i < n - 1; i++)

{

bool k = false;

for (int j = 0; j < n - i - 1; j++)

{

if (a[j] > a[j + 1])

{

swap(a[j], a[j + 1]);

k = true;

}

}

if (k == false)

break;

cout << "Buoc " << b++ << ": ";

for (int j = 0; j < n; j++)

cout << a[j] << " ";

cout << endl;

}

}

}

## CON6\_28 - Sắp xếp chèn

#include <bits/stdc++.h>

using namespace std;

int main(){

int t;

cin>>t;

while(t--){

int n;

cin >> n;

int b = n - 1;

vector<int> a(n);

stack<vector<int>> st;

for (int i = 0; i < n; i++)

cin >> a[i];

for (int i = 0; i < n; i++){

int k = a[i];

int j = i - 1;

while (j >= 0 && a[j] > k)

{

a[j + 1] = a[j];

j = j - 1;

}

a[j + 1] = k;

st.push(a);

}

while (st.size()){

a = st.top();

st.pop();

cout << "Buoc " << b-- << ": ";

for (int j = 0; j <= b + 1; j++)

cout << a[j] << " ";

cout << endl;

}

}

}

## CON7\_11 - BIẾN ĐỔI TRUNG TỐ - HẬU TỐ

#include <bits/stdc++.h>

#define endl "\n"

using namespace std;

string ts = "+-\*/^";

bool check(char c){

for (int i = 0; i < ts.size(); i++)

if (ts[i] == c)

return 1;

return 0;

}

bool Var(char c){

return ((c >= 'A' && c <= 'Z') || (c >= 'a' && c <= 'z'));

}

int Deg(char c){

if (c == '^')

return 50;

if (c == '\*' || c == '/')

return 100;

if (c == '+' || c == '-')

return 300;

return 1000;

}

int main(){

int t;

cin >> t;

while (t--){

string s, ans;

cin >> s;

stack<char> st;

st.push('#');

for (int i = 0; i < s.size(); i++){

if (s[i] == '('){

st.push(s[i]);

continue;

}

if (s[i] == ')'){

while (st.top() != '(')

{

ans += st.top();

st.pop();

}

st.pop();

continue;

}

if (Var(s[i])){

ans += s[i];

continue;

}

if (check(s[i])){

while (Deg(st.top()) <= Deg(s[i]))

{

ans += st.top();

st.pop();

}

st.push(s[i]);

continue;

}

}

while (st.top() != '#')

{

ans += st.top();

st.pop();

}

cout << ans << endl;

}

}

## CON7\_12 - BIẾN ĐỔI TIỀN TỐ - TRUNG TỐ

#include <bits/stdc++.h>

using namespace std;

string ts = "+-\*/";

bool check(char c){

for (int i = 0; i < ts.size(); i++)

if (ts[i] == c)

return 1;

return 0;

}

bool Var(char c){

return ((c >= 'a' && c <= 'z') || (c >= 'A' && c <= 'Z'));

}

int main(){

int t;

cin >> t;

while (t--){

string s, k;

cin >> s;

stack<string> st;

for (int i = s.size() - 1; i >= 0; i--){

if (check(s[i])){

k = '(' + st.top();

st.pop();

k += s[i];

k = k + st.top() + ')';

st.pop();

st.push(k);

}

else

st.push(string(1, s[i]));

}

cout << st.top() << endl;

}

}

## CON7\_13 - BIẾN ĐỐI TIỀN TỐ - HẬU TỐ

#include <bits/stdc++.h>

using namespace std;

string ts = "+-\*/";

bool check(char c){

for (int i = 0; i < ts.size(); i++)

if (ts[i] == c)

return 1;

return 0;

}

bool Var(char c){

return ((c >= 'a' && c <= 'z') || (c >= 'A' && c <= 'Z'));

}

int main(){

int t;

cin >> t;

while (t--){

string s, k1, k2;

cin >> s;

stack<string> st;

for (int i = s.size() - 1; i >= 0; i--){

if (check(s[i])){

k1 = st.top();

st.pop();

k2 = st.top();

st.pop();

st.push(k1 + k2 + s[i]);

}

else

st.push(string(1, s[i]));

}

cout << st.top() << endl;

}

}

## CON7\_15 - BIẾN ĐỔI HẬU TỐ - TRUNG TỐ

#include <bits/stdc++.h>

using namespace std;

bool Var(char c){

return ((c >= 'A' && c <= 'Z') || (c >= 'a' && c <= 'z'));

}

int main(){

int t;

cin >> t;

while (t--){

string s, k;

cin >> s;

stack<string> st;

for (int i = 0; i < s.size(); i++){

if (Var(s[i]))

st.push(string(1, s[i]));

else{

k = st.top();

st.pop();

k = s[i] + k;

k = st.top() + k;

st.pop();

k = "(" + k + ")";

st.push(k);

}

}

cout << st.top() << endl;

}

}

## CON7\_16 - TÍNH GIÁ TRỊ BIỂU THỨC HẬU TỐ

#include<bits/stdc++.h>

using namespace std;

long long tinh(long long a,long long b,char c){

if(c == '+') return a + b;

if(c == '-') return a - b;

if(c == '\*') return a \* b;

if(c == '/') return a / b;

}

long long tinh\_hau\_to(string s){

stack<long long> st;

for(int i = 0;i < s.size();i++){

if(s[i] == '+' || s[i] == '-' || s[i] == '\*' || s[i] == '/'){

long long a = st.top();st.pop();

long long b = st.top();st.pop();

long long tmp = tinh(b,a,s[i]);//phải lấy b trước để đảm bảo thứ tự

st.push(tmp);

}

else{

st.push((long long)(s[i] - '0'));

}

}

return st.top();

}

void solution(){

int t;

cin >> t;

while(t--){

string s;

cin >> s;

cout << tinh\_hau\_to(s) << endl;

}

}

int main(){

solution();

return 0;

}

## CON7\_17 - TÍNH GIÁ TRỊ BIỂU THỨC TIỀN TỐ

#include<bits/stdc++.h>

using namespace std;

long long tinh(long long a,long long b,char c){

if(c == '+') return a + b;

if(c == '-') return a - b;

if(c == '\*') return a \* b;

if(c == '/') return a / b;

}

long long tinh\_tien\_to(string s){

stack<long long> st;

for(int i = s.size() - 1;i >= 0;i--){

if(s[i] == '+' || s[i] == '-' || s[i] == '\*' || s[i] == '/'){

long long a = st.top();st.pop();

long long b = st.top();st.pop();

long long tmp = tinh(a,b,s[i]);

st.push(tmp);

}

else{

st.push((long long)(s[i] - '0'));

}

}

return st.top();

}

void solution(){

int t;

cin >> t;

while(t--){

string s;

cin >> s;

cout << tinh\_tien\_to(s) << endl;

}

}

int main(){

solution();

return 0;

}

## CON7\_18 - TÍNH TOÁN GIÁ TRỊ BIỂU THỨC TRUNG TỐ

#include <bits/stdc++.h>

using namespace std;

string op = "+-\*/";

bool check(char k)

{

for (int i = 0; i < op.size(); i++)

{

if (op[i] == k)

return 1;

}

return 0;

}

int deg(char k)

{

if (k == '\*' || k == '/')

return 10;

else if (k == '+' || k == '-')

return 20;

else

return 100;

}

string topost(string a)

{

string res = "";

stack<char> st;

st.push('#');

for (int i = 0; i < a.size(); i++)

{

if (a[i] == '(')

{

st.push(a[i]);

continue;

}

else if (a[i] == ')')

{

while (st.size() && st.top() != '(')

{

res += st.top();

st.pop();

}

st.pop();

continue;

}

else if (check(a[i]))

{

while (st.size() && deg(st.top()) <= deg(a[i]))

{

res += st.top();

st.pop();

}

st.push(a[i]);

continue;

}

long long so = 0, ok = 0;

while (i < a.size() && a[i] != ')' && !check(a[i]))

{

long long ss = a[i] - '0';

i++;

so = so \* 10 + ss;

ok = 1;

}

if (ok == 1)

i--;

res += "(";

res += to\_string(so);

res += ")";

}

while (st.size() && st.top() != '#')

{

res += st.top();

st.pop();

}

return res;

}

long long tinh(string a)

{

stack<long long> st;

int i = 0;

while (i < a.size())

{

if (a[i] == '(')

{

i++;

long long so = 0;

while (a[i] != ')')

{

so = so \* 10 + (a[i] - '0');

i++;

}

i++;

st.push(so);

}

else if (st.size() > 1)

{

long long tmp1 = st.top();

st.pop();

long long tmp2 = st.top();

st.pop();

if (a[i] == '+')

st.push(tmp1 + tmp2);

else if (a[i] == '-')

st.push(tmp2 - tmp1);

else if (a[i] == '\*')

st.push(tmp1 \* tmp2);

else if (a[i] == '/')

st.push(tmp2 / tmp1);

i++;

}

}

return st.top();

}

int main()

{

int t;

cin >> t;

while (t--)

{

string a;

cin >> a;

cout << tinh(topost(a)) << endl;

}

}

## CON7\_23 - GIẢI MÃ XÂU KÝ TỰ

#include <bits/stdc++.h>

using namespace std;

int main(){

int t;

cin >> t;

while (t--){

string s, k, ls, ans;

cin >> s;

int n;

stack<char> st;

for (int i = 0; i < s.size(); i++){

if (s[i] == ']')

{

k = ls = ans = "";

while (st.top() != '[')

{

k = st.top() + k;

st.pop();

}

st.pop();

while (st.size() && st.top() >= '0' && st.top() <= '9')

{

ls = st.top() + ls;

st.pop();

}

n = 0;

for (auto c : ls) n = n \* 10 + c - '0';

if (n == 0) n++;

for (int j = 0; j < n; j++) ans += k;

for (auto c : ans) st.push(c);

}

else st.push(s[i]);

}

ans = "";

while (st.size()){

ans = st.top() + ans;

st.pop();

}

cout << ans << endl;

}

}

## CON5\_2 - DÃY CON TĂNG DÀI NHẤT

#include <bits/stdc++.h>

using namespace std;

int main(){

int t;

cin >>t;

while(t--){

int n;

cin >>n;

int a[n+5], b[n+5];

for(int i = 1; i <= n; ++i) cin >>a[i];

int res = 1;

b[1] = a[1];

for(int i = 2; i <= n; ++i){

int x = lower\_bound(b + 1,res + b + 1, a[i]) - b;

b[x] = a[i];

res = max(res,x);

}

cout <<res <<'\n';

}

return 0;

}

## CON5\_3 - DÃY CON CÓ TỔNG BẰNG S

#include <iostream>

#include <vector>

using namespace std;

bool isSubsetSum(vector<int>& arr, int n, int sum) {

vector<vector<bool>> dp(n + 1, vector<bool>(sum + 1, false));

for (int i = 0; i <= n; i++) {

dp[i][0] = true;

}

for (int i = 1; i <= n; i++) {

for (int j = 1; j <= sum; j++) {

if (arr[i - 1] <= j) {

dp[i][j] = dp[i - 1][j] || dp[i - 1][j - arr[i - 1]];

} else {

dp[i][j] = dp[i - 1][j];

}

}

}

return dp[n][sum];

}

int main() {

int T;

cin >> T;

while (T--) {

int N, S;

cin >> N >> S;

vector<int> A(N);

for (int i = 0; i < N; i++) {

cin >> A[i];

}

if (isSubsetSum(A, N, S)) {

cout << "YES" << endl;

} else {

cout << "NO" << endl;

}

}

return 0;

}

## CON5\_11 - Số CATALAN

#include <bits/stdc++.h>

using namespace std;

int n, s[200], a[200], len;

void add(int x, int delta){

for (int i = 2; i \* i <= x; i++)

{

while (x % i == 0)

{

s[i] += delta;

x /= i;

}

}

if (x != 1)

s[x] += delta;

}

void mul(int x){

len += 5;

for (int i = 0; i < len; i++)

a[i] \*= x;

for (int i = 0; i < len; i++)

{

a[i + 1] += a[i] / 10;

a[i] %= 10;

}

while (len > 0 && !a[len - 1])

len--;

}

int main(){

ios\_base::sync\_with\_stdio(0);

cin.tie(0);

int t;

cin >> t;

while (t--){

cin >> n;

memset(a, 0, sizeof(a));

memset(s, 0, sizeof(s));

for (int i = 1; i <= n; i++)

add(2 \* n - i + 1, 1);

for (int i = 1; i <= n + 1; i++)

add(i, -1);

a[0] = len = 1;

for (int i = 1; i < 200; i++)

{

while (s[i]--)

mul(i);

}

for (int i = len - 1; i >= 0; i--)

cout << a[i];

cout << endl;

}

}

## CON5\_12 - TÍNH P(N,K)

#include <bits/stdc++.h>

using namespace std;

int main(){

long long P[1001][1001] = {}, mod = 1e9 + 7;

for (int i = 1; i <= 1000; i++)

P[i][1] = i;

for (int i = 2; i <= 1000; i++)

for (int j = 2; j <= i; j++)

P[i][j] = (P[i - 1][j - 1] \* i) % mod;

int t;

cin >> t;

while (t--)

{

int n, k;

cin >> n >> k;

cout << P[n][k] << endl;

}

}

## CON5\_13 - SỐ UGLY

#include <bits/stdc++.h>

using namespace std;

int main(){

set<long long> s;

vector<long long> v;

s.insert(1);

while (1){

for (auto i : s){

if (i > 1e18)

break;

v.push\_back(i \* 2);

v.push\_back(i \* 3);

v.push\_back(i \* 5);

}

for (auto i : v)

s.insert(i);

v.clear();

if (s.size() > 1e4)

break;

}

for (auto i : s)

v.push\_back(i);

int t;

cin >> t;

while (t--){

int n;

cin >> n;

cout << v[n - 1] << endl;

}

}

## CON5\_16 - TỔNG LỚN NHẤT CỦA DÃY CON TĂNG DẦN

#include <bits/stdc++.h>

using namespace std;

int main(){

int t;

cin >> t;

while (t--){

int n;

cin >> n;

int a[n], l[n], ans = 0;

for (int i = 0; i < n; i++){

cin >> a[i];

l[i] = a[i];

for (int j = 0; j < i; j++)

if (a[i] > a[j])

l[i] = max(l[i], l[j] + a[i]);

ans = max(ans, l[i]);

}

cout << ans << endl;

}

}

## CON5\_17 - DÃY SỐ BI-TONIC

#include <bits/stdc++.h>

using namespace std;

int main(){

int t;

cin >> t;

while (t--){

int n;

cin >> n;

int a[n], l[n], r[n], ans = 0;

for (int i = 0; i < n; i++){

cin >> a[i];

l[i] = a[i];

for (int j = 0; j < i; j++)

if (a[i] > a[j])

l[i] = max(l[i], l[j] + a[i]);

}

for (int i = n - 1; i >= 0; i--){

r[i] = a[i];

for (int j = i + 1; j < n; j++)

if (a[i] > a[j])

r[i] = max(r[i], r[j] + a[i]);

ans = max(ans, l[i] + r[i] - a[i]);

}

cout << ans << endl;

}

}

## CON5\_18 - Cặp số

#include <bits/stdc++.h>

using namespace std;

int main(){

int t;

cin >> t;

while (t--){

int n;

cin >> n;

int l[n], maz = 0;

vector<pair<int, int>> a(n);

for (int i = 0; i < n; i++)

cin >> a[i].first >> a[i].second;

sort(a.begin(), a.end());

for (int i = 0; i < n; i++){

l[i] = 1;

for (int j = 0; j < i; j++)

{

if (a[i].first > a[j].second && l[i] <= l[j])

l[i] = l[j] + 1;

}

maz = max(maz, l[i]);

}

cout << maz << endl;

}

}

## CON5\_19 - KÝ TỰ GIỐNG NHAU

#include <bits/stdc++.h>

using namespace std;

int main(){

int t;

cin >> t;

while (t--){

int n, x, y, z;

cin >> n >> x >> y >> z;

int a[n + 2];

a[1] = x;

a[2] = a[1] + min(x, z);

for (int i = 3; i <= n + 1; i++){

a[i] = a[i - 1] + x;

if (i % 2 == 0)

a[i] = min(a[i], a[i / 2] + z);

a[i - 1] = min(a[i - 1], a[i] + y);

}

cout << a[n] << endl;

}

}

## CON5\_20 - TỔNG CÁC XÂU CON

#include <bits/stdc++.h>

using namespace std;

int main(){

int t;

cin >> t;

while (t--){

string a;

cin >> a;

long long ans = 0, k = 1, n = a.size(), s;

for (int i = a.size() - 1; i >= 0; i--){

s = (long long)(a[i] - '0') \* k;

s \*= n;

ans += s;

k = k \* 10 + 1;

n--;

}

cout << ans << endl;

}

}

## CON5\_21 - TỔNG BẰNG K

#include <bits/stdc++.h>

using namespace std;

int main(){

int t;

cin >> t;

while (t--){

int n, k;

cin >> n >> k;

int a[n];

long long f[k + 1] = {}, mod = 1e9 + 7;

for (int i = 0; i < n; i++)

cin >> a[i];

f[0] = 1;

for (int i = 1; i <= k; i++)

for (int j = 0; j < n; j++)

if (i >= a[j])

f[i] = (f[i] + f[i - a[j]]) % mod;

cout << f[k] << endl;

}

}

## CON5\_22 - TỔNG LỚN NHẤT CỦA DÃY CON KHÔNG KỀ NHAU

#include <bits/stdc++.h>

using namespace std;

int main(){

int t;

cin >> t;

while (t--){

int n;

cin >> n;

int a[n + 2];

a[0] = a[1] = 0;

for (int i = 2; i <= n + 1; i++){

cin >> a[i];

a[i] = max(a[i] + a[i - 2], a[i - 1]);

}

cout << max(a[n], a[n + 1]) << endl;

}

}

## CON5\_23 - SỐ BƯỚC ÍT NHẤT

#include <bits/stdc++.h>

using namespace std;

int main(){

int t;

cin >> t;

while (t--){

int n, s = 0;

cin >> n;

int a[n], l[n];

for (int i = 0; i < n; i++){

cin >> a[i];

l[i] = 1;

for (int j = 0; j < i; j++){

if (a[i] >= a[j] && l[i] <= l[j])

l[i] = l[j] + 1;

}

s = max(s, l[i]);

}

cout << n - s << endl;

}

}

## CON5\_24 - DI CHUYỂN VỀ GỐC TỌA ĐỘ

#include <bits/stdc++.h>

using namespace std;

int main(){

int t;

cin >> t;

while (t--){

int n, m;

cin >> n >> m;

vector<vector<long long>> a(n + 1, vector<long long>(m + 1, 1));

for (int i = n - 1; i >= 0; i--)

for (int j = m - 1; j >= 0; j--)

a[i][j] = a[i + 1][j] + a[i][j + 1];

cout << a[0][0] << endl;

}

}

## CON5\_25 - Con ếch

#include <iostream>

using namespace std;

int countWays(int n) {

int dp[n + 1];

dp[0] = 1;

dp[1] = 1;

dp[2] = 2;

for (int i = 3; i <= n; i++) {

dp[i] = dp[i - 1] + dp[i - 2] + dp[i - 3];

}

return dp[n];

}

int main() {

int T;

cin >> T;

while (T--) {

int n;

cin >> n;

cout << countWays(n) << endl;

}

return 0;

}

## CON5\_27 - Cái Túi

#include <bits/stdc++.h>

using namespace std;

int main(){

int t;

cin >> t;

while (t--){

int n, m;

cin >> n >> m;

int a[n], b[n];

vector<vector<int>> l(n, vector<int>(m + 1, 0));

for (int i = 0; i < n; i++)

cin >> a[i];

for (int i = 0; i < n; i++)

cin >> b[i];

for (int i = 0; i < n; i++)

for (int j = 1; j <= m; j++){

if (i > 0){

if (a[i] <= j)

l[i][j] = max(l[i - 1][j], l[i - 1][j - a[i]] + b[i]);

else

l[i][j] = l[i - 1][j];

}

else if (a[i] <= j)

l[i][j] = b[i];

}

cout << l[n - 1][m] << endl;

}

}

## CON5\_29 - giải mã

#include <bits/stdc++.h>

using namespace std;

int main(){

int t;

cin >> t;

while (t--){

string s;

cin >> s;

s = ' ' + s;

int n = s.size(), k;

int a[n];

a[0] = a[1] = 1;

if (s[1] == '0'){

cout << 0 << endl;

continue;

}

for (int i = 2; i < n; i++){

if (s[i] == '0'){

if (s[i - 1] > '2')

{

a[n - 1] = 0;

break;

}

a[i] = 0;

}

else

a[i] = a[i - 1];

k = (s[i - 1] - '0') \* 10 + s[i] - '0';

if (k <= 26 && k >= 10)

a[i] += a[i - 2];

}

cout << a[n - 1] << endl;

}

}

## CON5\_30 - TỔNG BÌNH PHƯƠNG

#include <bits/stdc++.h>

using namespace std;

int main(){

int t;

cin >> t;

while (t--){

int n;

cin >> n;

int a[n + 1];

a[0] = 0;

for (int i = 1; i <= n; i++){

a[i] = 1e9;

for (int j = 1; j <= sqrt(i); j++)

if (a[i - j \* j] != 1e9)

a[i] = min(a[i], a[i - j \* j] + 1);

}

cout << a[n] << endl;

}

}

## CON5\_28 - Biến đổi xâu

#include <bits/stdc++.h>

using namespace std;

string s1, s2;

int dp(){

int m = s1.size(), n = s2.size();

int f[m + 1][n + 1];

for (int i = 0; i <= m; i++){

for (int j = 0; j <= n; j++){

if (i == 0)

f[i][j] = j;

else if (j == 0)

f[i][j] = i;

else if (s1[i - 1] == s2[j - 1])

f[i][j] = f[i - 1][j - 1];

else

f[i][j] = 1 + min(f[i][j - 1], min(f[i - 1][j], f[i - 1][j - 1]));

}

}

return f[m][n];

}

int main(){

int t;

cin >> t;

while (t--){

cin >> s1 >> s2;

cout << dp() << endl;

}

}

## CON5\_1 - XÂU CON CHUNG DÀI NHẤT

#include <bits/stdc++.h>

using namespace std;

int LCS(string s1, string s2){

int F[s1.size() + 1][s2.size() + 1] = {};

for (int i = 0; i < s1.size(); i++){

for (int j = 0; j < s2.size(); j++){

if (s1[i] == s2[j])

F[i + 1][j + 1] = F[i][j] + 1;

else

F[i + 1][j + 1] = max(F[i][j + 1], F[i + 1][j]);

}

}

return F[s1.size()][s2.size()];

}

int main(){

int t;

cin >> t;

while (t--){

string a, b;

cin >> a >> b;

cout << LCS(a, b) << endl;

}

}

## CON8\_01 - CẤU TRÚC DỮ LIỆU HÀNG ĐỢI 1

#include <bits/stdc++.h>

using namespace std;

int main(){

int t;

cin >> t;

while (t--){

int n, x;

cin >> n;

deque<int> st;

while (n--){

cin >> x;

if (x == 1)

cout << st.size() << endl;

else if (x == 2){

if (st.empty())

cout << "YES" << endl;

else

cout << "NO" << endl;

}

else if (x == 3){

cin >> x;

st.push\_back(x);

}

else if (x == 4){

if (st.size())

st.pop\_front();

}

else if (x == 5){

if (st.size())

cout << st.front() << endl;

else

cout << -1 << endl;

}

else if (x == 6){

if (st.size())

cout << st.back() << endl;

else

cout << -1 << endl;

}

}

}

}

## CON8\_02 - CẤU TRÚC DỮ LIỆU HÀNG ĐỢI 2

#include <bits/stdc++.h>

using namespace std;

int main(){

int n, x;

cin >> n;

string s;

deque<int> dq;

while (n--){

cin >> s;

if (s == "PUSH"){

cin >> x;

dq.push\_back(x);

}

else if (s == "PRINTFRONT"){

if (dq.size())

cout << dq.front() << endl;

else

cout << "NONE" << endl;

}

else if (s == "POP"){

if (dq.size())

dq.pop\_front();

}

}

}

## CON8\_4 - GIÁ TRỊ NHỎ NHẤT CỦA XÂU

#include <bits/stdc++.h>

using namespace std;

int main(){

int t;

cin >> t;

while (t--){

int k, x, n = 0;

string s;

cin >> k >> s;

map<char, int> m;

priority\_queue<int> q;

for (int i = 0; i < s.size(); i++)

m[s[i]]++;

for (auto i : m)

q.push(i.second);

while (k--){

x = q.top();

q.pop();

q.push(x - 1);

}

while (q.size()){

n += q.top() \* q.top();

q.pop();

}

cout << n << endl;

}

}

## CON8\_05 - SỐ NHỊ PHÂN TỪ 1 ĐẾN N

#include <bits/stdc++.h>

using namespace std;

string to\_binary(int n){

string s = "";

while (n > 0){

if (n % 2 == 0){

s = '0' + s;

}

else {

s = '1' + s;

n--;

}

n /= 2;

}

return s;

}

int main(){

int t;

cin >> t;

while (t--){

int n;

cin >> n;

for (int i = 1; i <= n; i++){

cout << to\_binary(i) << " ";

}

cout << endl;

}

}

## CON8\_06 - SỐ 0 VÀ SỐ 9

#include <bits/stdc++.h>

using namespace std;

int main(){;

int t;

cin >> t;

while (t--){

int n;

cin >> n;

queue<long long> q;

q.push(9);

while (q.size()){

long long k = q.front();

q.pop();

if (k % n == 0){

cout << k << endl;

break;

}

q.push(k \* 10);

q.push(k \* 10 + 9);

}

}

}

## CON8\_07 - SỐ BDN 1

#include <bits/stdc++.h>

using namespace std;

string n, k;

bool check(string s){

    return s.size() < n.size() || (s.size() == n.size() && s <= n);

}

int main(){

    int t;

    cin >> t;

    while (t--){

        int d = 0;

        cin >> n;

        stack<string> st;

        st.push("1");

        while (st.size()){

            k = st.top();

            d++;

            st.pop();

            if (check(k + "0"))

                st.push(k + "0");

            if (check(k + "1"))

                st.push(k + "1");

        }

        cout << d << endl;

    }

}

## CON8\_08 - SỐ BDN 2

#include <bits/stdc++.h>

using namespace std;

int main(){

int t;

cin >> t;

while (t--){

long long n, k;

cin >> n;

queue<long long> q;

q.push(1);

while (q.size()){

k = q.front();

if (k % n == 0){

cout << k << endl;

break;

}

q.pop();

q.push(k \* 10);

q.push(k \* 10 + 1);

}

}

}

## CON8\_09 - BIẾN ĐỔI S – T

#include <bits/stdc++.h>

using namespace std;

int main(){

int t;

cin >> t;

while (t--){

int s, t;

cin >> s >> t;

pair<int, int> u;

u.first = s;

u.second = 0;

map<int, int> m;

queue<pair<int, int>> q;

q.push(u);

while (q.size()){

u = q.front();

if (u.first == t){

cout << u.second << endl;

break;

}

q.pop();

if (u.first <= t && !m[u.first \* 2]){

m[u.first \* 2]++;

q.push({u.first \* 2, u.second + 1});

}

if (u.first > 1 && !m[u.first - 1]){

m[u.first - 1]++;

q.push({u.first - 1, u.second + 1});

}

}

}

}

## CON8\_10 - BIẾN ĐỔI SỐ TỰ NHIÊN

#include <bits/stdc++.h>

using namespace std;

int main(){

int t;

cin >> t;

while (t--){

int n;

cin >> n;

pair<int, int> u, v;

queue<pair<int, int>> q;

map<int, int> m;

u.first = n;

u.second = 0;

q.push(u);

while (q.size()){

u = q.front();

q.pop();

if (u.first == 2){

cout << u.second + 1 << endl;

break;

}

for (int i = 2; i <= sqrt(u.first); i++){

if (u.first % i == 0 && !m[u.first / i]){

m[u.first / i]++;

q.push({u.first / i, u.second + 1});

}

}

q.push({u.first - 1, u.second + 1});

}

}

}

## CON8\_11 - BIẾN ĐỔI SỐ NGUYÊN TỐ

#include <iostream>

#include <vector>

#include <queue>

#include <cmath>

using namespace std;

bool isPrime(int num) {

if (num <= 1) {

return false;

}

for (int i = 2; i <= sqrt(num); i++) {

if (num % i == 0) {

return false;

}

}

return true;

}

int minSteps(int S, int T) {

if (S == T) {

return 0;

}

vector<bool> visited(10000, false);

queue<pair<int, int>> q;

q.push({S, 0});

visited[S] = true;

while (!q.empty()) {

int curr = q.front().first;

int steps = q.front().second;

q.pop();

for (int i = 0; i < 4; i++) {

int digit = (curr / (int)pow(10, i)) % 10;

for (int j = 0; j <= 9; j++) {

int next = curr + (j - digit) \* pow(10, i);

if (next >= 1000 && next <= 9999 && !visited[next] && isPrime(next)) {

if (next == T) {

return steps + 1;

}

q.push({next, steps + 1});

visited[next] = true;

}

}

}

}

return -1;

}

int main() {

int T;

cin >> T;

while (T--) {

int S, T;

cin >> S >> T;

cout << minSteps(S, T) << endl;

}

return 0;

}

## CON8\_12 - KHOẢNG CÁCH XÂU KÝ TỰ

#include <bits/stdc++.h>

using namespace std;

int main(){;

int t;

cin >> t;

while (t--){

int n;

string st, ed, k;

cin >> n >> st >> ed;

set<string> s;

pair<string, int> u, v;

queue<pair<string, int>> q;

for (int i = 1; i <= n; i++){

cin >> k;

s.insert(k);

}

s.erase(st);

u.first = st;

u.second = 1;

q.push(u);

while (q.size()){

u = q.front();

q.pop();

if (u.first == ed){

cout << u.second << endl;

break;

}

v.second = u.second + 1;

for (int i = 0; i < u.first.size(); i++){

k = u.first;

for (char j = 'A'; j <= 'Z'; j++){

k[i] = j;

if (s.find(k) != s.end()){

s.erase(k);

v.first = k;

q.push(v);

}

}

}

}

}

}

## CON8\_13 - TÌM SỐ K THỎA MÃN ĐIỀU KIỆN

#include <bits/stdc++.h>

using namespace std;

vector<int> v;

int m[6];

void Try(int n){

v.push\_back(n);

for (int i = 0; i <= 5; i++){

if (!m[i]){

m[i] = 1;

Try(n \* 10 + i);

m[i] = 0;

}

}

}

int main(){

for (int i = 1; i <= 5; i++){

m[i] = 1;

Try(i);

m[i] = 0;

}

v.push\_back(0);

sort(v.begin(), v.end());

int t;

cin >> t;

while (t--){

int l, r, s = 0;

cin >> l >> r;

int k1 = lower\_bound(v.begin(), v.end(), l) - v.begin();

int k2 = lower\_bound(v.begin(), v.end(), r) - v.begin();

if (v[k2] != r)

k2--;

cout << k2 - k1 + 1 << endl;

}

}

## CON8\_15 - DI CHUYỂN TRONG MA TRẬN

#include <bits/stdc++.h>

using namespace std;

struct point{

int x, y, s;

};

int main(){

int t;

cin >> t;

while (t--){

int n, m, ans = 1e9;

cin >> n >> m;

vector<vector<bool>> check(n + 1, vector<bool>(m + 1, 1));

vector<vector<int>> a(n + 1, vector<int>(m + 1));

for (int i = 1; i <= n; i++)

for (int j = 1; j <= m; j++)

cin >> a[i][j];

point st, k;

queue<point> q;

st.x = st.y = 1;

st.s = 0;

q.push(st);

while (q.size()){

st = q.front();

q.pop();

if (st.x == n && st.y == m)

ans = min(ans, st.s);

k = st;

k.s++;

if (st.x + a[st.x][st.y] <= n && check[st.x + a[st.x][st.y]][st.y]){

check[st.x + a[st.x][st.y]][st.y] = 0;

k.x += a[st.x][st.y];

q.push(k);

k.x -= a[st.x][st.y];

}

if (st.y + a[st.x][st.y] <= m && check[st.x][st.y + a[st.x][st.y]]){

check[st.x][st.y + a[st.x][st.y]] = 0;

k.y += a[st.x][st.y];

q.push(k);

k.y -= a[st.x][st.y];

}

}

if (ans == 1e9)

cout << -1 << endl;

else

cout << ans << endl;

}

}

## CON8\_16 - Quay hình vuông

#include <bits/stdc++.h>

#define endl "\n"

using namespace std;

int moves[6][2] = {{3, 0}, {0, 4}, {2, 1}, {4, 3}, {1, 5}, {5, 2}};

string rleft(string a){

string b = "";

for (int i = 0; i < 6; i++)

b += a[moves[i][0]];

return b;

}

string rright(string a){

string b = "";

for (int i = 0; i < 6; i++)

b += a[moves[i][1]];

return b;

}

int main(){

int t;

cin >> t;

while (t--){

string a, b;

int x;

for (int i = 0; i < 6; i++){

cin >> x;

a += to\_string(x);

}

for (int i = 0; i < 6; i++){

cin >> x;

b += to\_string(x);

}

unordered\_set<string> visited;

queue<pair<string, int>> q;

q.push({a, 0});

while (!q.empty()){

string curr = q.front().first;

int steps = q.front().second;

q.pop();

if (curr == b){

cout << steps << endl;

break;

}

string next = rleft(curr);

if (visited.find(next) == visited.end()){

visited.insert(next);

q.push({next, steps + 1});

}

next = rright(curr);

if (visited.find(next) == visited.end()){

visited.insert(next);

q.push({next, steps + 1});

}

}

}

return 0;

}

## CON8\_18 - Gieo mầm

#include <bits/stdc++.h>

using namespace std;

int rx[] = {1, -1, 0, 0};

int ry[] = {0, 0, 1, -1};

struct data

{

int x, y, s;

};

int main(){

int t;

cin >> t;

while (t--){

int n, m, ans = 0;

cin >> n >> m;

int a[n + 1][m + 1];

struct data u, v;

queue<struct data> q;

u.s = 0;

for (int i = 1; i <= n; i++){

for (int j = 1; j <= m; j++){

cin >> a[i][j];

if (a[i][j] == 2){

u.x = i;

u.y = j;

q.push(u);

}

}

}

while (q.size()){

u = q.front();

q.pop();

ans = max(ans, u.s);

for (int i = 0; i < 4; i++){

v.x = u.x + rx[i];

v.y = u.y + ry[i];

if (v.x >= 1 && v.x <= n && v.y >= 1 && v.y <= m && a[v.x][v.y] == 1)

{

a[v.x][v.y] = 2;

v.s = u.s + 1;

q.push(v);

}

}

}

for (int i = 1; i <= n; i++)

for (int j = 1; j <= m; j++)

if (a[i][j] == 1)

ans = -1;

cout << ans << endl;

}

}

## CON8\_19 - DI CHUYỂN TRONG KHÔNG GIAN

#include <bits/stdc++.h>

using namespace std;

struct point

{

int x, y, z, s;

};

int gx[6] = {1, -1, 0, 0, 0, 0};

int gy[6] = {0, 0, 1, -1, 0, 0};

int gz[6] = {0, 0, 0, 0, 1, -1};

int main(){

int t;

cin >> t;

while (t--)

{

int h, r, d, ok = -1;

cin >> h >> r >> d;

point st, ed, b, c;

char a[h + 1][r + 1][d + 1];

queue<point> q;

for (int i = 1; i <= h; i++)

{

cin.ignore();

for (int j = 1; j <= r; j++)

{

for (int k = 1; k <= d; k++)

{

cin >> a[i][j][k];

if (a[i][j][k] == 'S')

{

st.x = i;

st.y = j;

st.z = k;

st.s = 0;

a[i][j][k] = '#';

}

else if (a[i][j][k] == 'E')

{

ed.x = i;

ed.y = j;

ed.z = k;

}

}

}

}

q.push(st);

while (q.size())

{

b = q.front();

q.pop();

c = b;

c.s++;

if (b.x == ed.x && b.y == ed.y && b.z == ed.z)

{

ok = b.s;

break;

}

for (int i = 0; i < 6; i++)

{

c.x += gx[i];

c.y += gy[i];

c.z += gz[i];

if (c.x >= 1 && c.x <= h && c.y >= 1 && c.y <= r && c.z >= 1 && c.z <= d && a[c.x][c.y][c.z] != '#')

{

q.push(c);

a[c.x][c.y][c.z] = '#';

}

c.x -= gx[i];

c.y -= gy[i];

c.z -= gz[i];

}

}

cout << ok << endl;

}

}

## CON8\_20 - HEXGAME

#include <bits/stdc++.h>

#define fi first

#define se second

using namespace std;

int l1[10] = {0, 4, 1, 3, 8, 5, 2, 7, 9, 6};

int r1[10] = {3, 0, 2, 7, 4, 1, 6, 8, 5, 9};

int l2[10] = {1, 5, 2, 0, 4, 8, 6, 3, 7, 9};

int r2[10] = {0, 2, 6, 3, 1, 5, 9, 7, 4, 8};

string left1(string a)

{

string b = "";

for (int i = 0; i < 10; i++)

b += a[l1[i]];

return b;

}

string right1(string a)

{

string b = "";

for (int i = 0; i < 10; i++)

b += a[r1[i]];

return b;

}

string left2(string a)

{

string b = "";

for (int i = 0; i < 10; i++)

b += a[l2[i]];

return b;

}

string right2(string a)

{

string b = "";

for (int i = 0; i < 10; i++)

b += a[r2[i]];

return b;

}

int main()

{

int t;

cin >> t;

while (t--)

{

string a, b, y;

int x, mim = 1e9;

for (int i = 0; i < 10; i++)

{

cin >> x;

a += to\_string(x);

}

b = "1238004765";

pair<string, int> x1, x2, y1, y2, k;

map<string, int> f;

x1.fi = a;

x1.se = 0;

x2.fi = b;

x2.se = 0;

queue<pair<string, int>> q1, q2;

q1.push(x1);

q2.push(x2);

while (q1.size())

{

y1 = q1.front();

q1.pop();

k.se = y1.se + 1;

k.fi = left1(y1.fi);

if (f[k.fi] == 0)

f[k.fi] = k.se;

q1.push(k);

k.fi = right1(y1.fi);

if (f[k.fi] == 0)

f[k.fi] = k.se;

q1.push(k);

if (k.se == 14)

break;

}

while (q2.size())

{

y2 = q2.front();

q2.pop();

if (f[y2.fi] > 0)

mim = min(mim, f[y2.fi] + y2.se);

k.se = y2.se + 1;

k.fi = left2(y2.fi);

q2.push(k);

k.fi = right2(y2.fi);

q2.push(k);

if (k.se == 14)

break;

}

cout << mim << endl;

}

}

## CON9\_02 - CHUYỂN TỪ DANH SÁCH KỀ SANG DANH SÁCH CẠNH

#include<bits/stdc++.h>

using namespace std;

int main(){

int n;

cin>>n;

vector<pair<int,int>>v;

int x;

char c;

for(int i=1;i<=n;i++){

while(scanf("%d%c",&x,&c)!=EOF){

if(x>i){

v.push\_back({i,x});

}

if(c=='\n'){

break;

}

}

}

for(auto it:v){

cout<<it.first<<" "<<it.second<<endl;

}

return 0;

}

## CON9\_03 - CHUYỂN MA TRẬN KỀ SANG DANH SÁCH KỀ

#include <iostream>

#include <vector>

using namespace std;

int main() {

int n;

cin >> n;

vector<vector<int>> adjacencyMatrix(n, vector<int>(n));

for (int i = 0; i < n; i++) {

for (int j = 0; j < n; j++) {

cin >> adjacencyMatrix[i][j];

}

}

vector<vector<int>> adjacencyList(n);

for (int i = 0; i < n; i++) {

for (int j = 0; j < n; j++) {

if (adjacencyMatrix[i][j] == 1) {

adjacencyList[i].push\_back(j + 1);

}

}

}

for (int i = 0; i < n; i++) {

for (int j = 0; j < adjacencyList[i].size(); j++) {

cout << adjacencyList[i][j] << " ";

}

cout << endl;

}

return 0;

}

## CON9\_04 - CHUYỂN DANH SÁCH KỀ SANG MA TRẬN KỀ

#include <bits/stdc++.h>

using namespace std;

int main(){

int n;

cin >> n;

string s;

vector<vector<int>> a(n + 1, vector<int>(n + 1, 0));

cin.ignore();

for (int i = 1; i <= n; i++)

{

getline(cin, s);

s += ' ';

int k = 0;

for (int j = 0; j < s.size(); j++)

{

if (s[j] >= '0' && s[j] <= '9')

k = k \* 10 + s[j] - '0';

else

{

a[i][k] = 1;

k = 0;

}

}

}

for (int i = 1; i <= n; i++)

{

for (int j = 1; j <= n; j++)

cout << a[i][j] << " ";

cout << endl;

}

return 0;

}

## CON9\_05 - BIỂU DIỄN ĐỒ THỊ CÓ HƯỚNG

#include<bits/stdc++.h>

using namespace std;

void xu\_ly(){

vector<int> arr[1005];

int n,m;

cin >> n >> m;

for(int i = 1;i <= m;i++){

int u,v;

cin >> u >> v;

arr[u].push\_back(v);

}

for(int i = 1;i <= n;i++){

cout << i << ": ";

for(int j = 0;j < arr[i].size();j++){

cout << arr[i][j] << " ";

}

cout << endl;

}

}

void solution(){

int t;

cin >> t;

while(t--){

xu\_ly();

}

}

int main(){

solution();

return 0;

}

## CON9\_07 - DFS TRÊN ĐỒ THỊ CÓ HƯỚNG

#include <bits/stdc++.h>

using namespace std;

int main(){

    int t;

    cin >> t;

    while (t--){

        int n, m, u, x, y;

        cin >> n >> m >> u;

        vector<bool> check(n + 1, 1);

        vector<vector<int>> a(n + 1);

        stack<int> st;

        while (m--)

        {

            cin >> x >> y;

            a[x].push\_back(y);

        }

        for (int i = 1; i <= n; i++)

            sort(a[i].begin(), a[i].end());

        st.push(u);

        check[u] = 0;

        cout << u << " ";

        while (st.size()){

            u = st.top();

            st.pop();

            for (int i = 0; i < a[u].size(); i++)

            {

                int v = a[u][i];

                if (check[v])

                {

                    cout << v << " ";

                    check[v] = 0;

                    st.push(u);

                    st.push(v);

                    break;

                }

            }

        }

        cout << endl;

    }

}

## CON9\_08 - BFS TRÊN ĐỒ THỊ VÔ HƯỚNG

#include <bits/stdc++.h>

using namespace std;

int main(){

    int t;

    cin >> t;

    while (t--)

    {

        int n, m, u, x, y;

        cin >> n >> m >> u;

        vector<int> a[n + 1], check(n + 1, 0);

        for (int i = 0; i < m; i++)

        {

            cin >> x >> y;

            a[x].push\_back(y);

            a[y].push\_back(x);

        }

        queue<int> q;

        q.push(u);

        check[u] = 1;

        while (q.size())

        {

            u = q.front();

            q.pop();

            cout << u << " ";

            for (int i = 0; i < a[u].size(); i++)

            {

                int v = a[u][i];

                if (!check[v])

                {

                    q.push(v);

                    check[v] = 1;

                }

            }

        }

        cout << endl;

    }

    return 0;

}

## CON9\_9 - BFS trên đồ thị có hướng

#include<bits/stdc++.h>

using namespace std;

vector<int> arr[1005];

int n,m,k;

bool check[1005];

void faster(){

ios\_base::sync\_with\_stdio(false);

cin.tie(NULL);

cout.tie(NULL);

}

void BFS(int u){

queue<int> q;

q.push(u);

check[u] = false;#include<bits/stdc++.h>

using namespace std;

vector<int> arr[1005];

int n,m,k;

bool check[1005];

void BFS(int u){

queue<int> q;

q.push(u);

check[u] = false;

while(!q.empty()){

int s = q.front();

cout << s << " ";

q.pop();

for(int i = 0;i < arr[s].size();i++){

if(check[arr[s][i]]){

q.push(arr[s][i]);

check[arr[s][i]] = false;

}

}

}

}

void solution(){

int t;

cin >> t;

while(t--){

memset(check,true,sizeof(check));

for(int i = 1;i <= n;i++){

arr[i].clear();

}

cin >> n >> m >> k;

for(int i = 1;i <= m;i++){

int u,v;

cin >> u >> v;

arr[u].push\_back(v);

}

BFS(k);

cout << endl;

}

}

int main(){

solution();

return 0;

}

while(!q.empty()){

int s = q.front();

cout << s << " ";

q.pop();

for(int i = 0;i < arr[s].size();i++){

if(check[arr[s][i]]){

q.push(arr[s][i]);

check[arr[s][i]] = false;

}

}

}

}

void solution(){

faster();

int t;

cin >> t;

while(t--){

memset(check,true,sizeof(check));

for(int i = 1;i <= n;i++){

arr[i].clear();

}

cin >> n >> m >> k;

for(int i = 1;i <= m;i++){

int u,v;

cin >> u >> v;

arr[u].push\_back(v);

}

BFS(k);

cout << endl;

}

}

int main(){

solution();

return 0;

}

## CON9\_10 - TÌM ĐƯỜNG ĐI THEO DFS VỚI ĐỒ THỊ VÔ HƯỚNG

#include<bits/stdc++.h>

using namespace std;

int n,m,s,t,check[1001];

int par[1001];

vector<int>adj[1001];

void init(){

cin>>n>>m>>s>>t;

memset(check,0,sizeof(check));

for(int i=0;i<1001;i++)adj[i].clear();

memset(par,0,sizeof(par));

for(int i=1;i<=m;i++){

int x,y;

cin>>x>>y;

adj[x].push\_back(y);

adj[y].push\_back(x);

}

}

void dfs(int u,int p){

check[u]=true;

for(int x:adj[u]){

if(!check[x]){

par[x]=u;

dfs(x,u);

}

}

}

void solve(){

dfs(s,0);

vector<int>path;

if(!check[t]){

cout<<"-1\n";

}

else{

while(t!=s){

path.push\_back(t);

t=par[t];

}

path.push\_back(s);

reverse(path.begin(),path.end());

for(int x:path)cout<<x<<" ";

cout<<"\n";

}

}

int main()

{

int tc;

cin >> tc;

while (tc--)

{

init();

solve();

}

return 0;

}

## CON9\_11 - TÌM ĐƯỜNG ĐI THEO DFS VỚI ĐỒ THỊ CÓ HƯỚNG

#include <bits/stdc++.h>

using namespace std;

int main()

{

int t;

cin >> t;

while (t--)

{

int n, m, u, v, x, y, ok = 0;

cin >> n >> m >> u >> v;

string s;

vector<bool> check(n + 1, 1);

vector<vector<int>> a(n + 1);

stack<pair<int, string>> st;

while (m--)

{

cin >> x >> y;

a[x].push\_back(y);

//a[y].push\_back(x);

}

st.push({u, to\_string(u) + " "});

check[u] = 0;

while (st.size())

{

x = st.top().first;

s = st.top().second;

if (x == v)

{

ok = 1;

cout << s << endl;

break;

}

st.pop();

for (int i = 0; i < a[x].size(); i++)

{

int y = a[x][i];

if (check[y])

{

check[y] = 0;

st.push({x, s});

st.push({y, s + to\_string(y) + " "});

break;

}

}

}

if (ok == 0)

cout << -1 << endl;

}

return 0;

}

## CON9\_12 - ĐƯỜNG ĐI THEO BFS TRÊN ĐỒ THỊ VÔ HƯỚNG

#include<bits/stdc++.h>

using namespace std;

int n,m,s,t,check[1001];

int par[1001];

vector<int>adj[1001];

void init(){

cin>>n>>m>>s>>t;

memset(check,0,sizeof(check));

for(int i=0;i<1001;i++)adj[i].clear();

memset(par,0,sizeof(par));

for(int i=1;i<=m;i++){

int x,y;

cin>>x>>y;

adj[x].push\_back(y);

adj[y].push\_back(x);

}

}

void bfs(int u,int p){

check[u]=true;

queue<int>q;

q.push(u);

while(!q.empty()){

int v=q.front();

q.pop();

for(int x:adj[v]){

if(!check[x]){

par[x]=v;

check[x]=true;

q.push(x);

}

}

}

}

void solve(){

bfs(s,0);

vector<int>path;

if(!check[t]){

cout<<"-1\n";

}

else{

while(t!=s){

path.push\_back(t);

t=par[t];

}

path.push\_back(s);

reverse(path.begin(),path.end());

for(int x:path)cout<<x<<" ";

cout<<"\n";

}

}

int main()

{

int tc;

cin >> tc;

while (tc--)

{

init();

solve();

}

return 0;

}

## CON9\_13 - ĐƯỜNG THI THEO BFS TRÊN ĐỒ THỊ CÓ HƯỚNG

#include<bits/stdc++.h>

using namespace std;

int n,m,s,t,check[1001];

int par[1001];

vector<int>adj[1001];

void init(){

cin>>n>>m>>s>>t;

memset(check,0,sizeof(check));

for(int i=0;i<1001;i++)adj[i].clear();

memset(par,0,sizeof(par));

for(int i=1;i<=m;i++){

int x,y;

cin>>x>>y;

adj[x].push\_back(y);

}

}

void bfs(int u,int p){

check[u]=true;

queue<int>q;

q.push(u);

while(!q.empty()){

int v=q.front();

q.pop();

for(int x:adj[v]){

if(!check[x]){

par[x]=v;

check[x]=true;

q.push(x);

}

}

}

}

void solve(){

bfs(s,0);

vector<int>path;

if(!check[t]){

cout<<"-1\n";

}

else{

while(t!=s){

path.push\_back(t);

t=par[t];

}

path.push\_back(s);

reverse(path.begin(),path.end());

for(int x:path)cout<<x<<" ";

cout<<"\n";

}

}

int main()

{

int tc;

cin >> tc;

while (tc--)

{

init();

solve();

}

return 0;

}

## CON9\_14 - KIỂM TRA ĐƯỜNG ĐI

#include<bits/stdc++.h>

using namespace std;

vector<int> arr[1005];

bool check[1005];

int n,m,t,s,truoc[1005];

void DFS(int u){

check[u] = false;

for(int i = 0;i < arr[u].size();i++){

if(check[arr[u][i]]){

truoc[arr[u][i]] = u;

DFS(arr[u][i]);

}

}

}

void solution(){

int l;

cin >> l;

while(l--){

memset(check,true,sizeof(check));

memset(truoc,0,sizeof(truoc));

for(int i = 1;i <= n;i++){

arr[i].clear();

}

cin >> n >> m;

for(int i = 1;i <= m;i++){

int u,v;

cin >> u >> v;

arr[u].push\_back(v);

arr[v].push\_back(u);

}

int q;

cin >> q;

while(q--){

memset(check,true,sizeof(check));

memset(truoc,0,sizeof(truoc));

int t,s;

cin >> t >> s;

DFS(t);

if(truoc[s] == 0){

cout << "NO" << endl;

}

else{

cout << "YES" << endl;

}

}

}

}

int main(){

solution();

return 0;

}

## CON9\_15 - ĐẾM SỐ THÀNH PHẦN LIÊN THÔNG VỚI DFS

#include<bits/stdc++.h>

using namespace std;

int n,m,check[1001],ans;

vector<int>adj[1001];

void init(){

cin>>n>>m;

ans=0;

memset(check,0,sizeof(check));

for(int i=0;i<1001;i++)adj[i].clear();

for(int i=1;i<=m;i++){

int x,y;

cin>>x>>y;

adj[x].push\_back(y);

adj[y].push\_back(x);

}

}

void dfs(int u){

check[u]=true;

for(int x:adj[u]){

if(!check[x]){

dfs(x);

}

}

}

int main()

{

int t;

cin >> t;

while (t--)

{

init();

for(int i=1;i<=n;i++){

if(!check[i]){

ans++;

dfs(i);

}

}

cout<<ans<<"\n";

}

return 0;

}

## CON9\_16 - TÌM SỐ THÀNH PHẦN LIÊN THÔNG VỚI BFS

#include<bits/stdc++.h>

using namespace std;

vector<int> arr[1005];

int n,m;

bool check[1005];

void init(){

cin >> n >> m;

for(int i = 1;i <= 1005;i++){

arr[i].clear();

}

for(int i = 1;i <= m;i++){

int u,v;

cin >> u >> v;

arr[u].push\_back(v);

arr[v].push\_back(u);

}

memset(check,true,sizeof(check));

}

void BFS(int u){

queue<int> q;

q.push(u);

check[u] = false;

while(!q.empty()){

int s = q.front();

q.pop();

for(int i = 0;i < arr[s].size();i++){

if(check[arr[s][i]]){

q.push(arr[s][i]);

check[arr[s][i]] = false;

}

}

}

}

void solution(){

int t;

cin >> t;

while(t--){

init();

int count = 0;

for(int i = 1;i <= n;i++){

if(check[i]){

count++;

BFS(i);

}

}

cout << count << endl;

}

}

int main(){

solution();

return 0;

}

## CON9\_26 - KIỂM TRA CHU TRÌNH SỬ DỤNG DISJOIN SET

#include <bits/stdc++.h>

using namespace std;

int n, m, x, y, ok;

vector<int> svs;

vector<bool> check;

vector<vector<int>> a;

void BFS(int k)

{

check[k] = 0;

for (auto i : a[k])

{

if (check[i])

{

svs[i] = k;

BFS(i);

}

else if (i != svs[k])

{

ok = 1;

return;

}

}

}

int main()

{

int t;

cin >> t;

while (t--)

{

cin >> n >> m;

ok = 0;

a.clear();

svs.clear();

check.clear();

a.resize(n + 1);

svs.resize(n + 1, -1);

check.resize(n + 1, 1);

for (int i = 0; i < m; i++)

{

cin >> x >> y;

a[x].push\_back(y);

a[y].push\_back(x);

}

for (int i = 1; i <= n; i++)

if (check[i] && !ok)

BFS(i);

if (ok)

cout << "YES" << endl;

else

cout << "NO" << endl;

}

}

## CON9\_30 - CHU TRÌNH EULER TRONG ĐỒ THỊ CÓ HƯỚNG

#include <iostream>

#include <vector>

#include <stack>

using namespace std;

void dfs(int node, vector<vector<int>>& graph, vector<bool>& visited, stack<int>& path) {

visited[node] = true;

for (int neighbor : graph[node]) {

if (!visited[neighbor]) {

dfs(neighbor, graph, visited, path);

}

}

path.push(node);

}

bool hasEulerCycle(int V, int E, vector<pair<int, int>>& edges) {

vector<vector<int>> graph(V + 1);

vector<bool> visited(V + 1, false);

stack<int> path;

for (auto edge : edges) {

int u = edge.first;

int v = edge.second;

graph[u].push\_back(v);

}

for (int i = 1; i <= V; i++) {

if (!visited[i]) {

dfs(i, graph, visited, path);

}

}

if (path.size() != E + 1) {

return false;

}

return true;

}

int main() {

int T;

cin >> T;

while (T--) {

int V, E;

cin >> V >> E;

vector<pair<int, int>> edges(E);

for (int i = 0; i < E; i++) {

cin >> edges[i].first >> edges[i].second;

}

bool hasCycle = hasEulerCycle(V, E, edges);

cout << (hasCycle ? 1 : 0) << endl;

}

return 0;

}

## CON9\_31 - KIỂM TRA ĐỒ THỊ CÓ PHẢI LÀ CÂY HAY KHÔNG

#include<bits/stdc++.h>

using namespace std;

typedef long long ll;

const int mxN = 1001;

vector<int> adj[mxN];

int V, E, n;

bool flag;

bool visited[mxN];

int sz[mxN];

int par[mxN];

void make\_set(){

for(int i = 1; i <= n; ++i){

sz[i] = 1;

par[i] = i;

}

}

int Find(int u){

if(u == par[u]) return u;

return par[u] = Find(par[u]);

}

bool Union(int x, int y){

x = Find(x);

y = Find(y);

if(x == y) return false;

if(sz[x] < sz[y]) swap(x,y);

sz[x] += y;

par[y] = x;

return true;

}

int main(){

ios\_base::sync\_with\_stdio(false);

cin.tie(0);

int t;

cin >> t;

while(t--){

flag = true;

cin >> n;

make\_set();

for(int i = 0; i < n - 1; ++i){

int u, v;

cin >> u >> v;

if(!Union(u,v)) flag = false;

}

if(flag)

cout << "YES";

else

cout << "NO";

cout << '\n';

}

return 0;

}

## CON9\_32 - SỐ LƯỢNG HÒN ĐẢO

#include<bits/stdc++.h>

using namespace std;

int n,m,ans,a[505][505];

int dx[8]= {1,0,-1,0,-1,-1,1,1};

int dy[8]= {0,1,0,-1,-1,1,-1,1};

void init(){

cin>>n>>m;

ans=0;

for(int i=1;i<=n;i++){

for(int j=1;j<=m;j++){

cin>>a[i][j];

}

}

}

void dfs(int i,int j){

a[i][j]=0;

for(int k=0;k<8;k++){

int i1=i+dx[k];

int j1=j+dy[k];

if(i1>=1&&i1<=n&&j1>=1&&j1<=m&&a[i1][j1]){

dfs(i1,j1);

}

}

}

int main()

{

int t;

cin >> t;

while (t--)

{

init();

for(int i=1;i<=n;i++){

for(int j=1;j<=m;j++){

if(a[i][j]){

ans++;

dfs(i,j);

}

}

}

cout<<ans<<"\n";

}

return 0;

}

## CON9\_06 - DFS TRÊN ĐỒ THỊ VÔ HƯỚNG

#include<bits/stdc++.h>

using namespace std;

vector<int> arr[1005];

bool check[1005];

void DFS(int u){

cout << u << " ";// duyệt đỉnh u

check[u] = false;

for(int i = 0;i < arr[u].size();i++){

if(check[arr[u][i]] == true){//tìm đỉnh kề chưa xét

DFS(arr[u][i]);

}

}

}

void solution(){

int t;

cin >> t;

while(t--){

memset(check,true,sizeof(check));

// khởi tạo lại arr[i] sau mỗi bước

for(int i = 0;i < 1005;i++){

arr[i].clear();

}

int n,m,k;

cin >> n >> m >> k;

for(int i = 0;i < m;i++){

int u,v;

cin >> u >> v;

arr[u].push\_back(v);

arr[v].push\_back(u);

}

DFS(k);

cout << endl;

}

}

int main(){

solution();

return 0;

}

## CON11\_5 - DUYỆT CÂY 3

#include<iostream>

#include<queue>

using namespace std;

struct node {

int data;

node\* pLeft;

node\* pRight;

node(int x) {

this->data = x;

pLeft = pRight = NULL;

}

};

typedef node\* tree;

void add\_Node(tree& T, int u, int v, char c) {

if (T == NULL) {

T = new node(u);

node\* p = new node(v);

if (c == 'L')T->pLeft = p;

else T->pRight = p;

}

else {

if (T->data == u) {

node\* p = new node(v);

if (c == 'L')T->pLeft = p;

else T->pRight = p;

}

else {

if (T->pLeft != NULL)add\_Node(T->pLeft, u, v, c);

if (T->pRight != NULL)add\_Node(T->pRight, u, v, c);

}

}

}

void Load(tree T) {

queue<tree> X;

X.push(T);

while (X.size()) {

tree tmp = X.front();

X.pop();

cout << tmp->data << " ";

if (tmp->pLeft != NULL)X.push(tmp->pLeft);

if (tmp->pRight != NULL)X.push(tmp->pRight);

}

}

int main() {

int t; cin >> t;

while (t--) {

int n; cin >> n;

tree T = NULL;

for (int i = 0; i < n; i++) {

int u, v;

char c;

cin >> u >> v >> c;

add\_Node(T, u, v, c);

}

Load(T);

cout << endl;

}

return 0;

}

## Con11\_7 - duyệt cây theo xoắn ốc

#include<bits/stdc++.h>

#define v(x) vector<x>

#define tree node\*

#define pb(a) push\_back(a)

#define pf(a) push\_front(a)

#define FOR(i,l,r) for(int i=l;i<r;i++)

#define FORX(i,l,r,x) for(int i=l;i<r;i+=x)

#define FORD(i,l,r) for(int i=l;i>=r;i--)

#define correct(x,y,n,m) 0<=(x)&&(x)<(n)&&0<=(y)&&(y)<(m)

#define cin(M,n) FOR(i,0,n)cin>>M[i]

#define cout(M,n) FOR(i,0,n)cout<<M[i]<<" "

#define rs(M,x) memset(M,x,sizeof(M))

#define reset() FOR(i, 0, 1001)A[i].clear(),check[i]=false

#define run() int t; cin >> t; while (t--)

#define pq(x ) priority\_queue<x>

#define neg\_pq(x) priority\_queue<x, vector<x>, greater<x>>

#define all(M) M.begin(),M.end()

using namespace std;

struct node {

int data;

node\* pLeft;

node\* pRight;

node(int x) {

this->data = x;

pLeft = pRight = NULL;

}

};

void Add(tree&T,int u,int v,char c) {

if (T == NULL) {

T = new node(u);

tree p = new node(v);

if (c == 'L')T->pLeft = p;

else T->pRight = p;

}

else {

if (T->data == u) {

tree p = new node(v);

if (c == 'L')T->pLeft = p;

else T->pRight = p;

}

else {

if (T->pLeft != NULL)Add(T->pLeft, u, v, c);

if (T->pRight != NULL)Add(T->pRight, u, v, c);

}

}

}

void Load(tree T) {

if (T == NULL)return;

stack<tree> S1, S2;

S1.push(T);

while (S1.size() || S2.size()) {

while (S1.size()) {

tree tmp = S1.top();

S1.pop();

cout << tmp->data << " ";

if (tmp->pRight != NULL)S2.push(tmp->pRight);

if (tmp->pLeft != NULL)S2.push(tmp->pLeft);

}

while (S2.size()) {

tree tmp = S2.top();

S2.pop();

cout << tmp->data << " ";

if (tmp->pLeft != NULL)S1.push(tmp->pLeft);

if (tmp->pRight != NULL)S1.push(tmp->pRight);

}

}

}

int main(){

run() {

int n; cin >> n;

tree T = NULL;

FOR(i, 0, n) {

int u, v; char c;

cin >> u >> v >> c;

Add(T, u, v, c);

}

Load(T);

cout << endl;

}

}

## CON11\_9 - KIỂM TRA NODE LÁ

#include<iostream>

#include<set>

using namespace std;

struct node {

int data;

node\* pLeft;

node\* pRight;

node(int x) {

this->data = x;

pLeft = pRight = NULL;

}

};

typedef node\* tree;

void add\_Node(tree& T, int u, int v, char c) {

if (T == NULL) {

T = new node(u);

tree p = new node(v);

if (c == 'L')T->pLeft = p;

else T->pRight = p;

}

else {

if (T->data == u) {

tree p = new node(v);

if (c == 'L')T->pLeft = p;

else T->pRight = p;

}

else {

if (T->pLeft != NULL)add\_Node(T->pLeft, u, v, c);

if (T->pRight != NULL)add\_Node(T->pRight, u, v, c);

}

}

}

set<int> X;

void Load(tree T, int count) {

if (T->pLeft == NULL && T->pRight == NULL) {

X.insert(count);

}

else {

if (T->pLeft != NULL)Load(T->pLeft, count + 1);

if (T->pRight != NULL)Load(T->pRight, count + 1);

}

}

int main() {

int t; cin >> t;

while (t--) {

X.clear();

int n; cin >> n;

tree T = NULL;

for (int i = 0; i < n; i++) {

int u, v;

char c;

cin >> u >> v >> c;

add\_Node(T, u, v, c);

}

Load(T,0);

if (X.size() > 1)cout << 0 << endl;

else cout << 1 << endl;

}

}

## CON1\_03 - HOÁN VỊ KẾ TIẾP

#include <bits/stdc++.h>

const int MOD=1e9+7;

using namespace std;

int main(){

    int t;

    cin >>t;

    while(t--){

        int n;

        cin >>n;

        int a[n];

        for(int i = 0; i < n; ++i) cin >>a[i];

        if(next\_permutation(a,a+n)) for(int i = 0; i < n; ++i) cout <<a[i] <<' ';

        else for(int i = 0; i < n; ++i) cout <<i+1 <<' ';

        cout <<'\n';

    }

   return 0;

}

## CON1\_04 - XÂU AB CÓ ĐỘ DÀI N

#include <bits/stdc++.h>

using namespace std;

int main()

{

int t;

cin >> t;

while (t--)

{

int n;

cin >> n;

string a;

for (int i = 0; i < n; i++)

a += 'A';

while (1)

{

cout << a << " ";

int ok = 0;

for (int i = n - 1; i >= 0; i--)

{

if (a[i] == 'A')

{

ok = 1;

a[i] = 'B';

for (int j = i + 1; j < n; j++)

a[j] = 'A';

break;

}

}

if (ok == 0)

break;

}

cout << endl;

}

}

## CON1\_09 - MÃ GRAY 1

#include <bits/stdc++.h>

#define endl "\n"

using namespace std;

int main()

{

int t;

cin >> t;

while (t--)

{

int n;

cin >> n;

vector<string> a;

a.push\_back("0");

a.push\_back("1");

for (int i = 2; i < (1 << n); i = i << 1)

{

for (int j = i - 1; j >= 0; j--)

a.push\_back(a[j]);

for (int j = 0; j < i; j++)

a[j] = "0" + a[j];

for (int j = i; j < 2 \* i; j++)

a[j] = "1" + a[j];

}

for (int i = 0; i < a.size(); i++)

cout << a[i] << " ";

cout << endl;

}

}

## CON1\_10 - MÃ GRAY 2

#include <bits/stdc++.h>

using namespace std;

int main(){

vector<vector<string>> ans;

vector<string> a;

a.push\_back("0");

a.push\_back("1");

ans.push\_back(a);

for (int i = 0; i < 9; i++){

a.clear();

for (int j = 0; j < ans[i].size(); j++)

{

a.push\_back('0' + ans[i][j]);

}

for (int j = ans[i].size() - 1; j >= 0; j--)

{

a.push\_back('1' + ans[i][j]);

}

ans.push\_back(a);

}

int t;

cin >> t;

while (t--){

string s;

cin >> s;

int k;

vector<int> b;

for (int i = 0; i < ans[s.length() - 1].size(); i++)

{

if (ans[s.length() - 1][i] == s)

{

k = i;

break;

}

}

while (k != 0)

{

b.push\_back(k % 2);

k /= 2;

}

for (int i = 0; i < s.length() - b.size(); i++)

cout << 0;

for (int i = b.size() - 1; i >= 0; i--)

cout << b[i];

cout << endl;

}

}

## CON1\_11 - MÃ GRAY 3

#include <bits/stdc++.h>

using namespace std;

int main()

{

vector<vector<string>> ans;

vector<string> a;

a.push\_back("0");

a.push\_back("1");

ans.push\_back(a);

for (int i = 0; i < 9; i++)

{

a.clear();

for (int j = 0; j < ans[i].size(); j++)

{

a.push\_back('0' + ans[i][j]);

}

for (int j = ans[i].size() - 1; j >= 0; j--)

{

a.push\_back('1' + ans[i][j]);

}

ans.push\_back(a);

}

int t;

cin >> t;

while (t--)

{

string s;

cin >> s;

long long k = 0;

for (int i = s.length() - 1; i >= 0; i--)

{

if (s[i] == '1')

{

k += (1 << (s.length() - 1 - i));

}

}

cout << ans[s.length() - 1][k] << endl;

}

}

## CON1\_12 - XÂU NHỊ PHÂN CÓ K BIT 1

#include <bits/stdc++.h>

using namespace std;

int main()

{

int t;

cin >> t;

while (t--)

{

int n, k;

cin >> n >> k;

int a[k];

for (int i = 0; i < k; i++)

a[i] = i;

stack<string> st;

while (1)

{

string s = "";

for (int i = 0; i < n; i++)

s += '0';

for (int i = 0; i < k; i++)

s[a[i]] = '1';

st.push(s);

int ok = 0;

for (int i = k - 1; i >= 0; i--)

{

if (a[i] != n - k + i)

{

ok = 1;

a[i]++;

for (int j = i + 1; j < k; j++)

a[j] = a[j - 1] + 1;

break;

}

}

if (ok == 0)

break;

}

while (st.size())

{

cout << st.top() << endl;

st.pop();

}

}

}

## CON1\_14 - TẬP QUÂN SỰ

#include <bits/stdc++.h>

using namespace std;

int main()

{

    int t;

    cin >> t;

    while (t--)

    {

        int n, k, s = 0, ok = 0;

        cin >> n >> k;

        int a[k + 1], b[k + 1];

        map<int, int> m;

        for (int i = 1; i <= k; i++)

        {

            cin >> a[i];

            b[i] = a[i];

            m[a[i]]++;

        }

        for (int i = k; i >= 1; i--)

        {

            if (a[i] != n - k + i)

            {

                ok = 1;

                a[i]++;

                for (int j = i + 1; j <= k; j++)

                    a[j] = a[j - 1] + 1;

                break;

            }

        }

        for (int i = 1; i <= k; i++)

        {

            if (m[a[i]] == 0)

                s++;

        }

        if (ok == 0)

            s = k;

        cout << s << endl;

    }

}

## CON1\_15 - HOÁN VỊ TIẾP THEO CỦA CHUỖI SỐ

#include <bits/stdc++.h>

using namespace std;

int main()

{

int t;

cin >> t;

while (t--)

{

int test;

string s;

cin >> test >> s;

int n = s.length();

int i = n - 2;

while (s[i + 1] <= s[i])

i--;

if (i < 0)

cout << test << " BIGGEST" << endl;

else

{

int k = n - 1;

while (s[i] >= s[k])

k--;

swap(s[i], s[k]);

sort(s.begin() + i + 1, s.end());

cout << test << ' ' << s << endl;

}

}

}

## CON1\_17 - TÌM BỘI SỐ

#include <bits/stdc++.h>

using namespace std;

int main(){

int t;

cin >> t;

while (t--)

{

int n;

cin >> n;

queue<long long> q;

q.push(9);

while (q.size())

{

long long k = q.front();

q.pop();

if (k % n == 0)

{

cout << k << endl;

break;

}

q.push(k \* 10);

q.push(k \* 10 + 9);

}

}

}

## CON1\_18 - MÁY ATM

#include<bits/stdc++.h>

using namespace std;

typedef long long ll;

int n, k;

vector<int> a;

int res;

void Try(int i = 0, int s = 0, int cnt = 0){

if(cnt >= res) return;

if(s == k){

res = min(res, cnt);

return;

}

if(i == n) return;

if(s + a[i] <= k)

Try(i + 1, s + a[i], cnt + 1);

Try(i + 1, s, cnt);

}

int main(){

ios\_base::sync\_with\_stdio(false);

cin.tie(0);

int t;

cin >> t;

while(t--){

a.clear();

cin >> n >> k;

a.resize(n);

for(int i = 0; i < n; ++i)

cin >> a[i];

res = n - 1;

sort(a.begin(), a.end(), greater<int>());

Try();

cout << (res == n - 1 ? -1 : res) << '\n';

}

return 0;

}

## CON2\_21 - HOÁN VỊ XÂU KÝ TỰ

#include <bits/stdc++.h>

using namespace std;

string s;

int a[1000], dd[1000] = {};

void Try(int n)

{

for (int i = 0; i < s.size(); i++)

{

if (!dd[i])

{

a[n] = i;

dd[i] = 1;

if (n == s.size() - 1)

{

for (int i = 0; i < s.size(); i++)

cout << s[a[i]];

cout << " ";

}

else

Try(n + 1);

dd[i] = 0;

}

}

}

int main()

{

int t;

cin >> t;

while (t--)

{

cin >> s;

Try(0);

cout << endl;

}

}

## CON2\_24 - DÃY CON TỔNG BẰNG K

#include <bits/stdc++.h>

using namespace std;

int n, k, a[101], b[101], ok;

void Try(int posa, int posb, int s)

{

for (int i = posa + 1; i <= n; i++)

{

if (s + a[i] == k)

{

ok = 1;

b[posb + 1] = a[i];

cout << "[" << b[0];

for (int j = 1; j <= posb + 1; j++)

cout << " " << b[j];

cout << "] ";

return;

}

else if (s + a[i] < k)

{

b[posb + 1] = a[i];

Try(i, posb + 1, s + a[i]);

}

}

}

int main()

{

int t;

cin >> t;

while (t--)

{

ok = 0;

cin >> n >> k;

for (int i = 1; i <= n; i++)

cin >> a[i];

sort(a + 1, a + n + 1);

Try(0, -1, 0);

if (ok == 0)

cout << -1 << endl;

cout << endl;

}

}

## CON2\_26 - ĐỔI CHỖ CÁC CHỮ SỐ

#include <bits/stdc++.h>

using namespace std;

int main()

{

int t;

cin >> t;

while (t--)

{

int n;

string s;

cin >> n >> s;

while (n--)

{

int l = 0, r = s.size() - 1, k = r;

while (l < r && s[l + 1] <= s[l])

l++;

if (l == r)

break;

while (l < r)

{

if (s[r] > s[k])

k = r;

r--;

}

l = 0;

while (l < k && s[l] >= s[k])

l++;

swap(s[l], s[k]);

}

cout << s << endl;

}

}

## CON2\_27 - CHIA MẢNG

#include <bits/stdc++.h>

using namespace std;

int n, k, sum;

bool ans = 0;

vector<int> a;

void Try(int s, int tmp)

{

if (ans)

return;

if (tmp == k)

{

ans = 1;

return;

}

for (int i = 0; i < n; i++)

{

if (s == sum)

Try(0, tmp + 1);

else if (s < sum)

Try(s + a[i], tmp);

else

return;

}

}

int main()

{

int t;

cin >> t;

while (t--)

{

cin >> n >> k;

sum = 0;

a.clear();

a.resize(n);

ans = 0;

for (int i = 0; i < n; i++)

{

cin >> a[i];

sum += a[i];

}

if (sum % k != 0)

cout << 0 << endl;

else

{

sum /= k;

Try(0, 0);

cout << ans << endl;

}

}

}

## CON2\_33 - SẮP XẾP QUÂN HẬU 1

#include <bits/stdc++.h>

using namespace std;

int n, ans, c1[50], c2[50], c3[50];

void Try(int x)

{

if (x == n + 1)

{

ans++;

return;

}

for (int i = 1; i <= n; i++)

{

if (c1[i] == 0 && c2[i + x] == 0 && c3[x - i + n] == 0)

{

c1[i] = c2[i + x] = c3[x - i + n] = 1;

Try(x + 1);

c1[i] = c2[i + x] = c3[x - i + n] = 0;

}

}

}

int main()

{

int t;

cin >> t;

while (t--)

{

ans = 0;

for (int i = 0; i < 20; i++)

c1[i] = c2[i] = c3[i] = 0;

cin >> n;

Try(1);

cout << ans << endl;

}

}

## CON2\_34 - SẮP XẾP QUÂN HẬU 2

#include <bits/stdc++.h>

using namespace std;

int n = 8, ans;

int c1[20], c2[20], c3[20], a[10][10];

void Try(int x, int s)

{

if (x == n + 1)

{

ans = max(ans, s);

return;

}

for (int i = 1; i <= n; i++)

{

if (c1[i] == 0 && c2[i + x] == 0 && c3[x - i + n] == 0)

{

c1[i] = c2[i + x] = c3[x - i + n] = 1;

Try(x + 1, s + a[x][i]);

c1[i] = c2[i + x] = c3[x - i + n] = 0;

}

}

}

int main()

{

int t;

cin >> t;

while (t--)

{

ans = 0;

for (int i = 0; i < 20; i++)

c1[i] = c2[i] = c3[i] = 0;

for (int i = 1; i <= 8; i++)

{

for (int j = 1; j <= 8; j++)

cin >> a[i][j];

}

Try(1, 0);

cout << ans << endl;

}

}

## CON2\_35 - TẬP HỢP

#include <bits/stdc++.h>

using namespace std;

int main()

{

while (1)

{

int n, k, s, d = 0;

cin >> n >> k >> s;

if (n < k)

{

cout << 0 << endl;

continue;

}

if (n == 0 && k == 0 && s == 0)

return 0;

int a[k + 1];

for (int i = 1; i <= k; i++)

a[i] = i;

while (1)

{

int x = 0;

for (int i = 1; i <= k; i++)

x += a[i];

if (x == s)

d++;

int ok = 0;

for (int i = k; i >= 1; i--)

{

if (a[i] != n - k + i)

{

ok = 1;

a[i]++;

for (int j = i + 1; j <= k; j++)

a[j] = a[j - 1] + 1;

break;

}

}

if (ok == 0)

break;

}

cout << d << endl;

}

}

## CON2\_36 - BIỂU THỨC TOÁN HỌC

#include <bits/stdc++.h>

using namespace std;

int main()

{

int t;

cin >> t;

while (t--)

{

string ans = "NO";

int a[10], b[10], c[10];

for (int i = 1; i <= 5; i++)

{

cin >> a[i];

b[i] = i;

}

do

{

c[1] = c[2] = c[3] = c[4] = 1;

while (1)

{

int ok = 0;

int x = a[b[1]];

for (int i = 1; i <= 4; i++)

{

if (c[i] == 1)

x = x + a[b[i + 1]];

else if (c[i] == 2)

x = x - a[b[i + 1]];

else

x = x \* a[b[i + 1]];

}

if (x == 23)

{

ans = "YES";

break;

}

for (int i = 4; i >= 1; i--)

{

if (c[i] != 3)

{

ok = 1;

c[i]++;

for (int j = i + 1; j <= 4; j++)

c[j] = 1;

break;

}

}

if (ok == 0)

break;

}

} while (next\_permutation(b + 1, b + 6));

cout << ans << endl;

}

}

## CON2\_37 - ĐƯỜNG ĐI DÀI NHẤT

#include <bits/stdc++.h>

using namespace std;

int n, m, x, y, ans;

bool check[20][20];

vector<vector<int>> a(20);

void Try(int pos, int k)

{

ans = max(ans, k);

for (auto i : a[pos])

{

if (!check[i][pos])

{

check[i][pos] = check[pos][i] = 1;

Try(i, k + 1);

check[i][pos] = check[pos][i] = 0;

}

}

}

int main()

{

int t;

cin >> t;

while (t--)

{

ans = 0;

cin >> n >> m;

for (int i = 0; i < 20; i++)

a[i].clear();

for (int i = 0; i < m; i++)

{

cin >> x >> y;

a[x].push\_back(y);

a[y].push\_back(x);

}

for (int i = 0; i < n; i++)

Try(i, 0);

cout << ans << endl;

}

}

## CON2\_40 - NGƯỜI DU LỊCH

#include <bits/stdc++.h>

using namespace std;

int n, a[20], b[20] = {};

long long s = 1e15, c[20][20], S = 0;

void Try(int x)

{

for (int i = 2; i <= n; i++)

{

if (b[i] == 0)

{

a[x] = i;

b[i] = 1;

S += c[a[x - 1]][i];

if (x == n && S + c[i][1] < s)

s = S + c[i][1];

else if (S < s && x < n)

Try(x + 1);

S -= c[a[x - 1]][i];

b[i] = 0;

}

}

}

int main()

{

a[1] = 1;

b[1] = 1;

cin >> n;

for (int i = 1; i <= n; i++)

{

for (int j = 1; j <= n; j++)

{

cin >> c[i][j];

}

}

if (n == 1)

{

cout << 0;

return 0;

}

Try(2);

cout << s;

}

## CON3\_1 - ĐỔI TIỀN

#include <bits/stdc++.h>

using namespace std;

int main()

{

int t;

cin >> t;

int a[100001] = {};

int b[10] = {1, 2, 5, 10, 20, 50, 100, 200, 500, 1000};

for (int i = 1; i <= 100000; i++)

{

a[i] = 1e9;

for (int j = 0; j < 10; j++)

{

if (i == b[j])

{

a[i] = 1;

break;

}

if (i - b[j] > 0 && a[i - b[j]] > 0)

a[i] = min(a[i], a[i - b[j]] + 1);

}

}

while (t--)

{

int n;

cin >> n;

cout << a[n] << endl;

}

}

## CON3\_10 - NỐI DÂY 1

#include <bits/stdc++.h>

using namespace std;

int main()

{

int t;

cin >> t;

while (t--)

{

int n, x;

cin >> n;

long long s = 0;

priority\_queue<int, vector<int>, greater<int>> q;

for (int i = 0; i < n; i++)

{

cin >> x;

q.push(x);

}

while (q.size() > 1)

{

int s1 = q.top();

q.pop();

int s2 = q.top();

q.pop();

int k = s1 + s2;

s += k;

q.push(k);

}

cout << s << endl;

}

}

## CON3\_11 - NỐI DÂY 2

#include <bits/stdc++.h>

using namespace std;

int main()

{

int t;

cin >> t;

while (t--)

{

int n;

cin >> n;

int x, s = 0, mod = 1e9 + 7;

priority\_queue<int, vector<int>, greater<int>> q;

for (int i = 0; i < n; i++)

{

cin >> x;

q.push(x);

}

while (q.size() > 1)

{

int s1 = q.top();

q.pop();

int s2 = q.top();

q.pop();

int k = (s1 + s2) % mod;

s = (s + k) % mod;

q.push(k);

}

cout << s << endl;

}

}

## CON3\_12 - SẮP ĐẶT XÂU KÝ TỰ 1

#include <bits/stdc++.h>

using namespace std;

int main()

{

int t;

cin >> t;

while (t--)

{

string s;

cin >> s;

int a[30] = {}, MAX = 0;

for (int i = 0; i < s.size(); i++)

a[s[i] - 'a']++;

for (int i = 0; i < 30; i++)

MAX = max(MAX, a[i]);

if (MAX \* 2 > s.size() + 1)

cout << -1 << endl;

else

cout << 1 << endl;

}

}

## CON3\_13 - SẮP ĐẶT XÂU KÝ TỰ 2

#include<bits/stdc++.h>

using namespace std;

typedef long long ll;

const ll mod = 1e9 + 7;

int main(){

int t;

cin >> t;

while(t--){

int k;

cin >> k;

string s;

cin >> s;

int n = s.length();

bool flag = true;

unordered\_map<char, int> cnt;

for(char c: s)

cnt[c]++;

for(auto p : cnt){

if(p.second > (n + 1) / k)

flag = false;

}

cout << (flag ? 1 : -1) << '\n';

}

return 0;

}

## CON3\_15 - MUA LƯƠNG THỰC

#include <bits/stdc++.h>

using namespace std;

int main()

{

int t;

cin >> t;

while (t--)

{

int n, s, m;

cin >> n >> s >> m;

if (s \* m > (s - s / 7) \* n)

cout << -1 << endl;

else

{

for (int i = 1; i <= s - s / 7; i++)

{

if (n \* i >= s \* m)

{

cout << i << '\n';

break;

}

}

}

}

}

## CON3\_16 - SỐ NHỎ NHẤT

#include <bits/stdc++.h>

using namespace std;

string a;

int s, n, d, ok;

void Try(int sn)

{

if (ok)

return;

int n = a.size();

if (n == d)

{

if (sn == s)

{

ok = 1;

cout << a << endl;

}

return;

}

if (s - sn < 0 || s - sn > (d - n) \* 9)

return;

for (int i = 0; i <= 9; i++)

{

a += to\_string(i);

Try(sn + i);

a.pop\_back();

}

}

int main()

{

int t;

cin >> t;

while (t--)

{

ok = 0;

cin >> s >> d;

for (int i = 1; i <= 9; i++)

{

if (!ok)

{

a = to\_string(i);

Try(i);

a.pop\_back();

}

}

if (!ok)

cout << -1 << endl;

}

}

## CON3\_17 - GIÁ TRỊ NHỎ NHẤT CỦA XÂU

#include<bits/stdc++.h>

using namespace std;

typedef long long ll;

const int mod = 1e9 + 7;

int main(){

int t;

cin >> t;

while(t--){

int k;

cin >> k;

string s;

cin >> s;

unordered\_map<char,int> mp;

for(char c: s)

mp[c]++;

priority\_queue<ll> q;

for(auto p : mp)

q.push(p.second);

ll res = 0;

while(k--){

ll top = q.top(); q.pop();

top--; q.push(top);

}

while(!q.empty()){

res += q.top() \* q.top(); q.pop();

}

cout << res << '\n';

}

return 0;

}

## CON3\_19 - PHÂN SỐ ĐƠN VỊ

#include <bits/stdc++.h>

using namespace std;

int main()

{

int t;

cin >> t;

while (t--)

{

long long a, b, n, m, x;

cin >> a >> b;

while (b % a != 0)

{

n = a;

m = b;

x = b / a;

cout << 1 << "/" << x + 1 << " + ";

a = (x + 1) \* n - m;

b = (x + 1) \* b;

}

cout << 1 << "/" << b / a;

cout << '\n';

}

}

## CON3\_20 - BIỂU THỨC ĐÚNG

#include <bits/stdc++.h>

using namespace std;

int main(){

int t;

cin >> t;

while (t--){

string s;

cin >> s;

stack<char> st;

int k = 0, ans = 0;

for (int i = 0; i < s.size(); i++){

if (st.size() == 0)

st.push(s[i]);

else if (st.top() == '[' && s[i] == ']'){

st.pop();

k += 2;

}

else if (st.top() == ']' && s[i] == '['){

ans += st.size() + k;

st.pop();

}

else if (s[i] == '[')

st.push(s[i]);

else if (st.top() == ']' && s[i] == ']')

st.push(s[i]);

if (st.size() == 0)

k = 0;

}

cout << ans << endl;

}

}

## CON3\_03 - TÌM MAX

#include <bits/stdc++.h>

using namespace std;

int main(){

int t;

cin >> t;

while (t--)

{

int n;

cin >> n;

long long a[n], s = 0, mod = 1e9 + 7;

for (int i = 0; i < n; i++)

cin >> a[i];

sort(a, a + n);

for (int i = 0; i < n; i++)

s = (s + a[i] \* i) % mod;

cout << s << endl;

}

}

## CON3\_02 - NHẦM CHỮ SỐ

#include <bits/stdc++.h>

using namespace std;

int to\_five(string s){

int n = 0;

for (int i = 0; i < s.size(); i++){

if (s[i] == '6')

s[i] = '5';

n = n \* 10 + s[i] - '0';

}

return n;

}

int to\_six(string s){

int n = 0;

for (int i = 0; i < s.size(); i++){

if (s[i] == '5')

s[i] = '6';

n = n \* 10 + s[i] - '0';

}

return n;

}

int main(){

string a, b;

cin >> a >> b;

cout << to\_five(a) + to\_five(b) << " " << to\_six(a) + to\_six(b);

}

## CON3\_04 - TỔNG NHỎ NHẤT

#include <bits/stdc++.h>

using namespace std;

int main(){

int t;

cin >> t;

while (t--){

int n;

cin >> n;

int a[n];

long long s1 = 0, s2 = 0;

for (int i = 0; i < n; i++)

cin >> a[i];

sort(a, a + n);

for (int i = 0; i < n; i += 2)

s1 = s1 \* 10 + a[i];

for (int i = 1; i < n; i += 2)

s2 = s2 \* 10 + a[i];

cout << s1 + s2 << endl;

}

}

## CON3\_06 - SẮP XẾP THAM LAM

#include <bits/stdc++.h>

using namespace std;

int main(){

int t;

cin >> t;

while (t--){

int n;

cin >> n;

int a[n], b[n], ok = 1;

for (int i = 0; i < n; i++){

cin >> a[i];

b[i] = a[i];

}

sort(b, b + n);

for (int i = 0; i < n; i++){

if (a[i] != b[i] && a[i] != b[n - i - 1]){

ok = 0;

break;

}

}

if (ok == 1)

cout << "Yes" << endl;

else

cout << "No" << endl;

}

}

## CON3\_07 - GIÁ TRỊ NHỎ NHẤT CỦA BIỂU THỨC

#include <bits/stdc++.h>

using namespace std;

bool cmp(long long a, long long b){

return a > b;

}

int main(){

int t;

cin >> t;

while (t--)

{

int n;

cin >> n;

long long s = 0;

vector<long long> a(n), b(n);

for (int i = 0; i < n; i++)

cin >> a[i];

for (int i = 0; i < n; i++)

cin >> b[i];

sort(a.begin(), a.end());

sort(b.begin(), b.end(), cmp);

for (int i = 0; i < n; i++)

s += a[i] \* b[i];

cout << s << endl;

}

}

## CON3\_08 - SẮP XẾP CÔNG VIỆC 1

#include <iostream>

#include <vector>

#include <algorithm>

int findMaxActions(std::vector<int>& startTimes, std::vector<int>& endTimes) {

int n = startTimes.size();

std::vector<std::pair<int, int>> actions;

for (int i = 0; i < n; i++) {

actions.push\_back({startTimes[i], endTimes[i]});

}

std::sort(actions.begin(), actions.end(), [](const std::pair<int, int>& a, const std::pair<int, int>& b) {

return a.second < b.second;

});

int count = 1;

int endTime = actions[0].second;

for (int i = 1; i < n; i++) {

if (actions[i].first >= endTime) {

count++;

endTime = actions[i].second;

}

}

return count;

}

int main() {

int t;

std::cin >> t;

while (t--) {

int n;

std::cin >> n;

std::vector<int> startTimes(n);

std::vector<int> endTimes(n);

for (int i = 0; i < n; i++) {

std::cin >> startTimes[i];

}

for (int i = 0; i < n; i++) {

std::cin >> endTimes[i];

}

int maxActions = findMaxActions(startTimes, endTimes);

std::cout << maxActions << std::endl;

}

return 0;

}

## CON3\_09 - SẮP XẾP CÔNG VIỆC 2

#include <bits/stdc++.h>

using namespace std;

int main()

{

int t;

cin >> t;

while (t--)

{

int n, x, d = 0, ans = 0;

cin >> n;

int check[n + 1] = {};

vector<pair<int, int>> a(n + 1);

for (int i = 1; i <= n; i++)

cin >> x >> a[i].second >> a[i].first;

sort(a.begin() + 1, a.end(), greater<pair<int, int>>());

for (int i = 1; i <= n; i++)

{

for (int j = min(n, a[i].second); j >= 1; j--)

{

if (!check[j])

{

ans += a[i].first;

d++;

check[j] = 1;

break;

}

}

}

cout << d << " " << ans << endl;

}

}

## CON3\_14 - SỐ KHỐI LẬP PHƯƠNG

#include<bits/stdc++.h>

using namespace std;

typedef long long ll;

unordered\_set<ll> st;

string S;

ll res = -1;

void Try(string s = "", int i = 0){

if(i == S.length()){

if(s.length() > 0){

if(st.find(stoll(s)) != st.end()){

res = max(res, stoll(s));

}

}

return;

}

Try(s + S[i], i + 1);

Try(s, i + 1);

}

int main(){

int t;

cin >> t;

for(ll i = 1; i <= 100; ++i){

st.insert(i \* i \* i);

}

while(t--){

res = -1;

cin >> S;

Try();

cout << res << '\n';

}

return 0;

}

## CON3\_18 - SỐ MAY MẮN

#include <bits/stdc++.h>

using namespace std;

int main(){

int t;

cin >> t;

while (t--)

{

int n, ok = 0;

cin >> n;

for (int i = 0; i <= n / 4; i++)

{

if ((n - i \* 4) % 7 == 0)

{

ok = 1;

for (int j = 0; j < i; j++)

cout << 4;

for (int j = 0; j < (n - i \* 4) / 7; j++)

cout << 7;

cout << endl;

break;

}

}

if (ok == 0)

cout << -1 << endl;

}

}

## CON2\_19 - DÃY SỐ 1

#include <bits/stdc++.h>

using namespace std;

int main(){

int t;

cin >> t;

while (t--){

int n;

cin >> n;

vector<int> a(n), b(n);

for (int i = 0; i < n; i++)

cin >> a[i];

for (int i = 0; i < n; i++){

cout << "[" << a[0];

for (int j = 1; j < n - i; j++){

cout << " " << a[j];

b[j - 1] = a[j - 1] + a[j];

}

cout << "]" << endl;

for (int j = 0; j < n - i; j++){

a[j] = b[j];

}

}

}

}

## CON2\_20 - DÃY SỐ 2

#include <bits/stdc++.h>

using namespace std;

int main(){

int t;

cin >> t;

while (t--)

{

int n;

cin >> n;

string s = "";

vector<int> a(n), b(n);

stack<string> st;

for (int i = 0; i < n; i++)

cin >> a[i];

for (int i = 0; i < n; i++)

{

s += "[" + to\_string(a[0]);

for (int j = 1; j < n - i; j++)

{

s += " " + to\_string(a[j]);

b[j - 1] = a[j - 1] + a[j];

}

s += "]";

st.push(s);

s = "";

for (int j = 0; j < n - i; j++)

{

a[j] = b[j];

}

}

while (st.size())

{

cout << st.top() << " ";

st.pop();

}

cout << endl;

}

}

## CON2\_25 - TẬP CON BẰNG NHAU

#include <bits/stdc++.h>

using namespace std;

int n, s;

int a[105];

string ans;

void Try(int pos, int k)

{

if (ans == "YES")

return;

if (2 \* k == s)

{

ans = "YES";

return;

}

if (pos < n)

{

if (2 \* (k + a[pos]) <= s)

Try(pos + 1, k + a[pos]);

Try(pos + 1, k);

}

}

int main()

{

int t;

cin >> t;

while (t--)

{

s = 0;

ans = "NO";

cin >> n;

for (int i = 1; i <= n; i++)

{

cin >> a[i];

s += a[i];

}

if (s % 2 == 0)

Try(0, 0);

cout << ans << endl;

}

}

## CON2\_28 - TỔ HỢP SỐ CÓ TỔNG BẰNG X

#include <bits/stdc++.h>

using namespace std;

struct data

{

vector<int> b;

};

int n, k, a[25], c[105];

vector<struct data> d;

void Try(int x, int i, int s)

{

for (int ii = i; ii <= n; ii++)

{

s += a[ii];

c[x] = a[ii];

if (s == k)

{

struct data e;

for (int jj = 1; jj <= x; jj++)

e.b.push\_back(c[jj]);

d.push\_back(e);

}

else if (s < k)

Try(x + 1, ii, s);

s -= a[ii];

}

}

int main()

{

int t;

cin >> t;

while (t--)

{

d.clear();

cin >> n >> k;

for (int i = 1; i <= n; i++)

cin >> a[i];

sort(a + 1, a + n + 1);

Try(1, 1, 0);

if (d.size() == 0)

{

cout << "-1" << endl;

continue;

}

for (int i = 0; i < d.size(); i++)

{

cout << "[" << d[i].b[0];

for (int j = 1; j < d[i].b.size(); j++)

{

if (d[i].b[j] == 0)

break;

cout << " " << d[i].b[j];

}

cout << "]";

}

cout << endl;

}

}

## CON2\_31 - TỪ ĐIỂN

#include <bits/stdc++.h>

using namespace std;

int k, n, m, ok;

char b[4][4];

bool check[4][4];

vector<string> a;

map<string, int> c;

vector<pair<int, int>> p = {{-1, 0}, {0, -1}, {0, 1}, {1, 0}, {-1, -1}, {-1, 1}, {1, -1}, {1, 1}};

void Try(int i1, int i2, string x)

{

if (c[x] > 0)

{

cout << x << " ";

ok = 1;

}

for (int i = 0; i < 8; i++)

{

int p1 = i1 + p[i].first;

int p2 = i2 + p[i].second;

if (p1 < n && p1 >= 0 && p2 < m && p2 >= 0 && !check[p1][p2])

{

check[p1][p2] = 1;

Try(p1, p2, x + b[p1][p2]);

check[p1][p2] = 0;

}

}

}

int main(){

int t;

cin >> t;

while (t--) {

string s = "";

ok = 0;

c.clear();

cin >> k >> n >> m;

a.resize(k);

for (int i = 0; i < k; i++)

{

cin >> a[i];

c[a[i]]++;

}

sort(a.begin(), a.end());

for (int i = 0; i < n; i++)

for (int j = 0; j < m; j++)

cin >> b[i][j];

for (int i = 0; i < n; i++)

for (int j = 0; j < m; j++)

{

check[i][j] = 1;

Try(i, j, s + b[i][j]);

check[i][j] = 0;

}

if (ok == 0)

cout << -1;

cout << endl;

}

}

## CON2\_29 - DI CHUYỂN TRONG MA TRẬN

#include <bits/stdc++.h>

using namespace std;

int main()

{

int t;

cin >> t;

while (t--)

{

int n, m, x;

cin >> n >> m;

int a[n + 1][m + 1] = {};

a[0][1] = 1;

for (int i = 1; i <= n; i++)

{

for (int j = 1; j <= m; j++)

{

cin >> x;

a[i][j] = a[i - 1][j] + a[i][j - 1];

}

}

cout << a[n][m] << endl;

}

}

## CON2\_30 - SỐ NGUYÊN TỐ

#include <bits/stdc++.h>

using namespace std;

int n, m, p, ss;

vector<int> a, x;

vector<bool> check(205, 1);

vector<vector<int>> ans;

void era()

{

for (int i = 2; i <= 200; i++)

{

if (check[i])

{

a.push\_back(i);

for (int j = i \* i; j <= 200; j += i)

check[j] = 0;

}

}

m = a.size() - 1;

}

void Try(int pos, int k, int s)

{

for (int i = pos + 1; i <= m; i++)

{

if (s + a[i] == ss && k == n - 1)

{

x.push\_back(a[i]);

ans.push\_back(x);

x.pop\_back();

}

else if (k < n - 1 && s + a[i] < ss)

{

x.push\_back(a[i]);

Try(i, k + 1, s + a[i]);

x.pop\_back();

}

}

}

int main(){

era();

int t;

cin >> t;

while (t--)

{

ans.clear();

cin >> n >> p >> ss;

int b = 0;

while (a[b] <= p)

b++;

Try(b - 1, 0, 0);

cout << ans.size() << endl;

for (int i = 0; i < ans.size(); i++)

{

for (int j = 0; j < ans[i].size(); j++)

cout << ans[i][j] << " ";

cout << endl;

}

}

}

## CON2\_32 - LOẠI BỎ DẤU NGOẶC

#include <bits/stdc++.h>

using namespace std;

int k;

string s;

map<string, int> m;

bool check(string a)

{

if (a.size() < 2)

return 0;

stack<int> st;

for (int i = 0; i < a.size(); i++)

{

if (a[i] == '(')

st.push(1);

else if (a[i] == ')')

{

if (st.empty())

return 0;

else

st.pop();

}

}

if (st.empty())

return 1;

return 0;

}

void Try(string a, int pos)

{

if (check(a))

{

if (a.size() > k)

{

k = a.size();

m.clear();

m[a]++;

}

else if (a.size() == k)

m[a]++;

return;

}

for (int i = pos + 1; i < a.size(); i++)

{

if (a[i] == '(' || a[i] == ')')

{

string x = a;

x.erase(x.begin() + i, x.begin() + i + 1);

Try(x, i - 1);

}

}

}

int main(){

int t;

cin >> t;

while (t--)

{

k = 0;

m.clear();

cin >> s;

Try(s, -1);

if (m.empty())

cout << -1;

else

{

for (auto i : m)

cout << i.first << " ";

}

cout << endl;

}

}

## CON1\_06 - SINH HOÁN VỊ

#include <iostream>

#include <vector>

using namespace std;

void generatePermutations(vector<int>& current, vector<bool>& used, int n) {

if (current.size() == n) {

for (int i = 0; i < n; i++) {

cout << current[i];

}

cout << " ";

return;

}

for (int i = 1; i <= n; i++) {

if (!used[i]) {

current.push\_back(i);

used[i] = true;

generatePermutations(current, used, n);

current.pop\_back();

used[i] = false;

}

}

}

int main() {

int T;

cin >> T;

for (int t = 0; t < T; t++) {

int n;

cin >> n;

vector<int> current;

vector<bool> used(n + 1, false);

generatePermutations(current, used, n);

cout << endl;

}

return 0;

}

CÂY KHUNG CỦA ĐỒ THỊ THEO THUẬT TOÁN DFS

#include<bits/stdc++.h>

using namespace std;

#define fi first

#define se second

#define pb push\_back

#define sz size()

inline void DFS(int x, vector<vector<int>> &a, bitset<1005> &bs, vector<pair<int, int>> &vp)

{

    bs[x] = 1;

    for(int &i : a[x])

    {

        if(!bs[i])

        {

            vp.pb({x, i});

            DFS(i, a, bs, vp);

        }

    }

}

int main()

{

    int t = 1, e, v, u, x, y;

    cin >> t;

    while(t--)

    {

        cin >> v >> e >> u;

        vector<vector<int>> a(v + 5);

        bitset<1005> bs;

        vector<pair<int, int>> res;

        while(e--)

        {

            cin >> x >> y;

            a[x].pb(y);

            a[y].pb(x);

        }

        DFS(u, a, bs, res);

        if(res.sz == v - 1)

            for(pair<int, int> &i : res)

                cout << i.fi << ' ' << i.se << endl;

        else cout << -1 << endl;

    }

    return 0;

}

Dãy con chung dài nhất

#include <iostream>

#include <vector>

#include <algorithm>

using namespace std;

int longestCommonSubsequence(vector<int>& X, vector<int>& Y, vector<int>& Z, int nX, int nY, int nZ) {

    // Tạo mảng 3 chiều để lưu trữ kết quả

    vector<vector<vector<int>>> dp(nX + 1, vector<vector<int>>(nY + 1, vector<int>(nZ + 1, 0)));

    // Xây dựng mảng dp

    for (int i = 1; i <= nX; i++) {

        for (int j = 1; j <= nY; j++) {

            for (int k = 1; k <= nZ; k++) {

                if (X[i - 1] == Y[j - 1] && Y[j - 1] == Z[k - 1]) {

                    dp[i][j][k] = dp[i - 1][j - 1][k - 1] + 1;

                } else {

                    dp[i][j][k] = max(max(dp[i - 1][j][k], dp[i][j - 1][k]), dp[i][j][k - 1]);

                }

            }

        }

    }

    // Trả về độ dài dãy con chung dài nhất

    return dp[nX][nY][nZ];

}

int main() {

    int t;

    cin >> t;

    while (t--) {

        int nX, nY, nZ;

        cin >> nX;

        vector<int> X(nX);

        for (int i = 0; i < nX; i++) {

            cin >> X[i];

        }

        cin >> nY;

        vector<int> Y(nY);

        for (int i = 0; i < nY; i++) {

            cin >> Y[i];

        }

        cin >> nZ;

        vector<int> Z(nZ);

        for (int i = 0; i < nZ; i++) {

            cin >> Z[i];

        }

        int result = longestCommonSubsequence(X, Y, Z, nX, nY, nZ);

        cout << result << endl;

    }

    return 0;

}

**CÂY KHUNG CỦA ĐỒ THỊ THEO THUẬT TOÁN BFS**

#include<bits/stdc++.h>

using namespace std;

#define mp make\_pair

#define fi first

#define se second

#define pb push\_back

#define sz size()

#define ll long long

#define FOR(i, a, b) for(int i = a; i <= b; i++)

#define FORD(i, a, b) for(int i = a; i >= b; i--)

#define F(i, a, b) for(int i = a; i < b; ++i)

#define FD(i, a, b) for(int i = a; i > b; --i)

#define faster() ios\_base::sync\_with\_stdio(0); cin.tie(NULL);cout.tie(NULL);

#define vi vector<int>

#define vll vector<ll>

#define all(x) (x).begin(), (x).end()

#define endl '\n'

inline void BFS(int x, vector<vector<int>> &a, bitset<1005> &bs, vector<pair<int, int>> &vp)

{

    queue<int> q;

    bs[x] = 1;

    int u;

    q.push(x);

    while(q.sz)

    {

        u = q.front();

        q.pop();

        for(int &i : a[u])

        {

            if(!bs[i])

            {

                bs[i] = 1;

                q.push(i);

                vp.push\_back({u, i});

            }

        }

    }

}

int main()

{

    faster();

    int t = 1, e, v, u, x, y;

    cin >> t;

    while(t--)

    {

        cin >> v >> e >> u;

        vector<vector<int>> a(v + 5);

        bitset<1005> bs;

        vector<pair<int, int>> res;

        while(e--)

        {

            cin >> x >> y;

            a[x].pb(y);

            a[y].pb(x);

        }

        BFS(u, a, bs, res);

        if(res.sz == v - 1)

            for(pair<int, int> &i : res)

                cout << i.fi << ' ' << i.se << endl;

        else cout << -1 << endl;

    }

    return 0;

}

**Tổng các chữ số**

#include <iostream>

#include <cmath>

using namespace std;

long long sumDigits(long long x) {

    if (x < 10) return x \* (x + 1) / 2;

    int d = log10(x);

    long long \*a = new long long[d + 1];

    a[0] = 0;

    a[1] = 45;

    for (int i = 2; i <= d; i++)

        a[i] = a[i - 1] \* 10 + 45 \* pow(10, i - 1);

    long long p = pow(10, d);

    long long msd = x / p;

    long long result = msd \* a[d] + (msd \* (msd - 1) / 2) \* p + msd \* (1 + x % p) + sumDigits(x % p);

    delete[] a;

    return result;

}

long long sumDigitsInRange(long long n, long long m) {

    return sumDigits(m) - sumDigits(n - 1);

}

int main() {

    int t;

    cin >> t;

    while (t--) {

        long long n, m;

        cin >> n >> m;

        cout << sumDigitsInRange(n, m) << endl;

    }

    return 0;

}

**Biểu diễn công thức hóa học**

#include <iostream>

#include <map>

#include <stack>

#include <string>

#include <cctype>

#include <algorithm>

using namespace std;

void parseFormula(const string& formula, map<string, int>& atomCounts) {

    stack<map<string, int>> stk;

    stk.push({});

    int n = formula.size();

    for (int i = 0; i < n;) {

        if (isupper(formula[i])) {

            string atom;

            atom += formula[i++];

            while (i < n && islower(formula[i])) {

                atom += formula[i++];

            }

            int count = 0;

            while (i < n && isdigit(formula[i])) {

                count = count \* 10 + (formula[i++] - '0');

            }

            if (count == 0) count = 1;

            stk.top()[atom] += count;

        } else if (formula[i] == '(') {

            stk.push({});

            i++;

        } else if (formula[i] == ')') {

            map<string, int> top = stk.top();

            stk.pop();

            i++;

            int count = 0;

            while (i < n && isdigit(formula[i])) {

                count = count \* 10 + (formula[i++] - '0');

            }

            if (count == 0) count = 1;

            for (auto& [atom, cnt] : top) {

                stk.top()[atom] += cnt \* count;

            }

        }

    }

    for (auto& [atom, cnt] : stk.top()) {

        atomCounts[atom] += cnt;

    }

}

string countOfAtoms(const string& formula) {

    map<string, int> atomCounts;

    parseFormula(formula, atomCounts);

    string result;

    for (auto& [atom, count] : atomCounts) {

        result += atom;

        if (count > 1) {

            result += to\_string(count);

        }

    }

    return result;

}

int main() {

    int t;

    cin >> t;

    while (t--) {

        string formula;

        cin >> formula;

        cout << countOfAtoms(formula) << endl;

    }

    return 0;

}

**Tổng các chữ số của giai thừa**

#include <iostream>

#include <vector>

using namespace std;

int tinhTongChuSo(int n) {

    vector<int> ketQua(10000, 0);

    ketQua[0] = 1;

    int tong = 0;

    for (int i = 2; i <= n; i++) {

        for (int j = 0; j < ketQua.size(); j++) {

            ketQua[j] \*= i;

        }

        for (int j = 0; j < ketQua.size() - 1; j++) {

            ketQua[j + 1] += ketQua[j] / 10;

            ketQua[j] %= 10;

        }

    }

    for (int i = ketQua.size() - 1; i >= 0; i--) {

        if (ketQua[i] != 0) {

            for (int j = i; j >= 0; j--) {

                tong += ketQua[j];

            }

            break;

        }

    }

    return tong;

}

int main() {

    int t;

    cin >> t;

    while (t--) {

        int n;

        cin >> n;

        cout << tinhTongChuSo(n) << endl;

    }

    return 0;

}

**Cửa sổ trượt tối đa**

#include <iostream>

#include <deque>

#include <vector>

std::vector<int> maxSlidingWindow(std::vector<int>& nums, int k) {

    std::deque<int> dq;

    std::vector<int> result;

    for (int i = 0; i < nums.size(); i++) {

        // Loại bỏ các phần tử ngoài cửa sổ trượt

        while (!dq.empty() && dq.front() < i - k + 1) {

            dq.pop\_front();

        }

        // Loại bỏ các phần tử nhỏ hơn phần tử hiện tại

        while (!dq.empty() && nums[dq.back()] < nums[i]) {

            dq.pop\_back();

        }

        // Thêm phần tử hiện tại vào deque

        dq.push\_back(i);

        // Thêm phần tử lớn nhất vào kết quả

        if (i >= k - 1) {

            result.push\_back(nums[dq.front()]);

        }

    }

    return result;

}

int main() {

    int t;

    std::cin >> t;

    while (t--) {

        int n, k;

        std::cin >> n >> k;

        std::vector<int> nums;

        int temp;

        while (std::cin >> temp) {

            nums.push\_back(temp);

            if (std::cin.peek() == '\n') break;

        }

        std::vector<int> result = maxSlidingWindow(nums, k);

        for (int i = 0; i < result.size(); i++) {

            std::cout << result[i] << " ";

        }

        std::cout << std::endl;

    }

    return 0;

}

**Liệt kê cạnh cầu**

#include <iostream>

#include <vector>

#include <algorithm>

using namespace std;

const int MAX = 1001;

vector<int> adj[MAX];

bool visited[MAX];

int disc[MAX], low[MAX], parent[MAX];

vector<pair<int, int>> bridge;

void DFS(int u, int time) {

    visited[u] = true;

    disc[u] = low[u] = time++;

    for (int v : adj[u]) {

        if (!visited[v]) {

            parent[v] = u;

            DFS(v, time);

            low[u] = min(low[u], low[v]);

            if (low[v] > disc[u]) {

                bridge.push\_back({u, v});

            }

        } else if (v != parent[u]) {

            low[u] = min(low[u], disc[v]);

        }

    }

}

void findBridges(int V) {

    for (int i = 1; i <= V; i++) {

        visited[i] = false;

        disc[i] = low[i] = -1;

        parent[i] = -1;

    }

    bridge.clear();

    int time = 0;

    for (int i = 1; i <= V; i++) {

        if (!visited[i]) {

            DFS(i, time);

        }

    }

}

void printBridges() {

    if (bridge.empty()) {

        cout << "NO" << endl;

    } else {

        for (auto& edge : bridge) {

            if (edge.first > edge.second) {

                swap(edge.first, edge.second);

            }

        }

        sort(bridge.begin(), bridge.end());

        for (auto& edge : bridge) {

            cout << edge.first << " " << edge.second << " ";

        }

        cout << endl;

    }

}

int main() {

    int T;

    cin >> T;

    while (T--) {

        int V, E;

        cin >> V >> E;

        for (int i = 1; i <= V; i++) {

            adj[i].clear();

        }

        for (int i = 0; i < E; i++) {

            int u, v;

            cin >> u >> v;

            adj[u].push\_back(v);

            adj[v].push\_back(u);

        }

        findBridges(V);

        printBridges();

    }

    return 0;

}